AD NUMBER

ADB206445

NEW LIMITATION CHANGE

TO

Approved for public release, distribution unlimited

FROM

Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; Dec 79. Other requests shall be referred to AFWAL/MLTC, Wright-Patterson AFB, OH 45433.

AUTHORITY

Air Force Research Lab., Wright-Patterson AFB, OH ltr., dtd March 27, 2001.

AFWAL-TR-80-4115

ICAM MANUFACTURING COST/DESIGN GUIDE VOLUME III: COMPUTERIZATION

C. R. CLAYDON, M. LARSON, B. R. NOTON
BATTELLE'S COLUMBUS LABORATORIES
505 KING AVENUE
COLUMBUS, OHIO 43201

SEPTEMBER 1980

FINAL REPORT
SEPTEMBER 1977 — JULY 1979

Distribution ~~Limited~~ auth to U.S. Government Agencies Only; Test and Evaluation; Statement Applied December, 1979. Other Requests for This Document Must be Referred to AFWAL/MLTC, Wright-Patterson Air Force Base, Ohio 45433.

SUBJECT TO EXPORT CONTROL LAWS

This document contains information for manufacturing or using munitions of war. Export of the information contained herein, or release to foreign nationals within the United States, without first obtaining an export license, is a violation of the International Traffic in Arms Regulations. Such violation is subject to a penalty of up to 2 years imprisonment and a fine of $100,000 under 22 USC 2778.

Include this notice with any reproduced portion of this document.

MATERIALS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AFB, OH 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This technical report has been reviewed and is approved for publication.

Steven R. LeClair

STEVEN R. LE CLAIR, CAPTAIN, USAF
Project Manager

FOR THE COMMANDER

N G Tupper

NATHAN G. TUPPER
Chief
Computer Integrated Manufacturing Branch
Manufacturing Technology Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/MLTC, W-PAFB, OH 45433 to help us maintain a current mailing list".

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

The following notice applies to any unclassified (including originally classified and now declassified) technical reports released to "qualified U.S. contractors" under the provisions of DoD Directive 5230.25, Withholding of Unclassified Technical Data From Public Disclosure.

NOTICE TO ACCOMPANY THE DISSEMINATION OF EXPORT-CONTROLLED TECHNICAL DATA

1. Export of information contained herein, which includes, in some circumstances, release to foreign nationals within the United States, without first obtaining approval or license from the Department of State for items controlled by the International Traffic in Arms Regulations (ITAR), or the Department of Commerce for items controlled by the Export Administration Regulations (EAR), may constitute a violation of law.

2. Under 22 U.S.C. 2778 the penalty for unlawful export of items or information controlled under the ITAR is up to two years imprisonment, or a fine of $100,000, or both. Under 50 U.S.C., Appendix 2410, the penalty for unlawful export of items or information controlled under the EAR is a fine of up to $1,000,000, or five times the value of the exports, whichever is greater; or for an individual, imprisonment of up to 10 years, or a fine of up to $250,000, or both.

3. In accordance with your certification that establishes you as a "qualified U.S. Contractor", unauthorized dissemination of this information is prohibited and may result in disqualification as a qualified U.S. contractor, and may be considered in determining your eligibility for future contracts with the Department of Defense.

4. The U.S. Government assumes no liability for direct patent infringement, or contributory patent infringement or misuse of technical data.

5. The U.S. Government does not warrant the adequacy, accuracy, currency, or completeness of the technical data.

6. The U.S. Government assumes no liability for loss, damage, or injury resulting from manufacture or use for any purpose of any product, article, system, or material involving reliance upon any or all technical data furnished in response to the request for technical data.

7. If the technical data furnished by the Government will be used for commercial manufacturing or other profit potential, a license for such use may be necessary. Any payments made in support of the request for data do not include or involve any license rights.

8. A copy of this notice shall be provided with any partial or complete reproduction of these data that are provided to qualified U.S. contractors.

D E S T R U C T I O N   N O T I C E

For classified documents, follow the procedures in DoD 5200.22-M, Industrial Security Manual, Section II-19 or DoD 5200.1-R, Information Security Program Regulation, Chapter IX. For unclassified, limited documents, destroy by any method that will prevent disclosure of contents or reconstruction of the document.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
AFWAL-TR-80-4115		

4. TITLE (and Subtitle)
ICAM "MANUFACTURING COST/DESIGN GUIDE" (MC/DG)
VOLUME III: COMPUTERIZATION

5. TYPE OF REPORT & PERIOD COVERED
Final Report
September, 1977 - July, 1979

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
C. R. Claydon
M. Larson
B. R. Noton

8. CONTRACT OR GRANT NUMBER(s)
F33615-77-C-5027

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Battelle's Columbus Laboratories
505 King Avenue
Columbus, Ohio 43201

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Project No. 870-7

11. CONTROLLING OFFICE NAME AND ADDRESS
Materials Laboratory
Air Force Wright Aeronautical Laboratories
Air Force Systems Command
Wright-Patterson Air Force Base, Ohio 45433

12. REPORT DATE
September, 1980

13. NUMBER OF PAGES
226

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Distribution ~~Limited~~ Auth to U.S. Government only (test and evaluation of military hardware). Other requests for this document must be referred to Materials Laboratory (AFWAL/MLTC), Wright-Patterson AFB, Ohio 45433.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

DTIC QUALITY INSPECTED 2

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Aided Manufacturing	Trade Studies	Assemblies
Manufacturing Cost	Sheet Metal Parts	Mechanical Fastening
Design-to-Cost	Sheet Metal Forming	Part Count
Airframe Design	Aluminum Sheet	Advanced Composites
Fuselage Panels	Titanium Sheet	Graphite/Epoxy
Cost Drivers	Steel Sheet	Laminating

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The potential advantages of a computerized MC/DG system, as compared to hard copy guides, lie in its ability to utilize state-of-the-art data management and graphics display techniques to provide the user easy access to frequently updated data for use in manufacturing cost/structural performance trade-off studies. This program was undertaken to construct a sample system for validating the concept of a computerized MC/DG utilizing data submitted by

several aerospace companies. Through this concept validation effort, an implementation plan was evolved which would outline a full-scale MC/DG.

Concept validation was undertaken by determining the user needs and system requirements as a basis for system design. A sample data base was then generated for test evaluation and critique of the developed sample system. The sample system was constructed utilizing existing modules of the BASIS (Battelle Automated Search Information System) and the IGS (Integrated Graphic System) software.

In Phase I, manufacturing man-hour data for 20 aluminum, steel, and titanium sheet metal discrete parts made by different manufacturing methods were obtained from participating aerospace companies covering manufacturing costs for lot sizes of 5, 10, 25, and 50 parts. Both general and detailed ground rules were established before data evaluation was made to minimize data variations between companies. In Phase II, the data base was expanded to include consideration of advanced composites and mechanically fastened assemblies.

In order to demonstrate the value of a computerized system, three sample user sessions were developed which demonstrate the ability to retrieve data, tabulate the retrieved data, and display data for interactive consideration. Procedures are outlined for showing how the user can obtain qualitative cost-driver effect (CDE) data and illustrations through tabular and graphic display of the effect of lot size and part length on part cost. Due to data and graphic software package limitations, testing of data display formats in Phase II was limited. However, samples of usage for considering mechanically fastened part recurring costs through a plot of cost per fastener as affected by the number of fasteners per assembly were demonstrated.

A logical conclusion to the concept validation activity is a discussion of other ways in which the user might utilize a dynamic computerized system for assessing such factors as the impact of material price fluctuations, labor rate fluctuations, lot release size, plus the historical value of a computerized MC/DG in that past design trade-off data could be reevaluated to insure that the best possible part configurations have been selected. These discussions all center around the necessity for a dynamic flexible system that can accommodate changing user needs.

Considerations of a full-scale implementation plan for a computerized MC/DG were largely based on the experience gained in the previous concept validation study. The development of a system design and the characteristics of the data base contained therein is again based upon the needs of the user (who is primarily the aerospace designer). General specifications for hardware, software, and data maintenance procedures are discussed and the importance of a concise and highly readable user's guide is emphasized. The concluding portions of this report discuss MC/DG requirements for a generalized data base management system, the method of selecting a DBMS, and a synopsis of five major DBMS system.

PREFACE

This technical report covers the work performed under Contract No. F33615-77-C-5027, from September 19, 1977, through July 19, 1979, by Battelle's Columbus Laboratories (BCL) for the Air Force Wright Aernonautical Laboratories, Computer Integrated Manufacturing Branch (AFWAL/MLTC), Wright-Patterson Air Force Base, Ohio 45433.

1. USAF TECHNICAL DIRECTION

This program was administered under the technical direction of Capt. Dan L. Shunk, MLTC. Mr. David Judson, MLTC, was responsible for the for the MC/DG Computerization, BCL.

2. BCL TECHNICAL DIRECTION

The Program Manager of the MC/DG Data Development and Computerization Program was Mr. Bryan R. Noton, Design/Manufacturing Interaction Project Office, BCL.

Dr. Charles R. Claydon was the Task Leader for MC/DG Computerization supported by Mrs. Molly Larson, Information Systems, Modeling and Applied Statistics Section.

Accesion For	
NTIS CRA&I	☐
DTIC TAB	☒
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and / or Special
B-3	

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS
(Continued)

LIST OF TABLES

SECTION I

BACKGROUND

The concept for developing a computerized MC/DG has evolved during the current and previous MC/DG contracts and is based on the following considerations:

- Aerospace designers will be the primary users of the MC/DG.
- A computerized system will be used in performing manufacturing cost/structural performance trade-offs on alternative design configurations.
- It will support the user in selecting appropriate manufacturing processes, man-hour (cost) data, displayed in the desired formats, to conduct trade-offs between alternative design configurations.
- This tool should be a real time, interactive mode system designed to utilize state-of-the-art data management and graphics display techniques and the state-of-the-art computer resources.
- It should be implemented using standard languages and structure techniques to develop modular subsystems suitable for installation on computers now utilized by the aerospace industry and to provide for the transition to future hardware and software systems.

SECTION II

OBJECTIVES

The computerization task, in Contract No. F33615-77-C-5027, was essentially a concept validation study that serves as an example of how the final, full-scale computerized MC/DG system could perform. Therefore, the prime objectives of this concept validation study were:

- Short range--the construction of the sample system for concept validation; this also serves as an example of how individual aerospace companies might construct a computerized MC/DG system from presently available computer software components and integrate it into company design manufacturing systems and information systems.
- Long range--the development of an implementation plan for a full-scale computerized MC/DG system which would be available for an aerospace company to install on its host computer.

SECTION III

APPROACH

This report is organized in two parts, each part consisting of several chapters as follows:

- Concept Validation for a Computerized MC/DG. This part describes the design, file structure, maintenance procedures, and key modules of the sample system constructed to validate the computerized MC/DG concept. The sections on functional system requirements are independent of particular software modules that could be integrated to support the required system functions. In the sections of this report on the Sample Computerized MC/DG and the Samples of Usage, a particular choice of existing software modules was made for the purpose of concept validation.
- Implementation Plan for a Computerized MC/DG. This part describes a proposed implementation plan for a full-scale computerized MC/DG, system interface with a general database management system, and interface with a current state-of-the-art computer system.

The computerization task was accomplished by performance of four subtasks:

(1) Gather, organize, and process data for computer storage, retrieval, and display

(2) Approve format design and implementation for computer storage, retrieval, and display

(3) Construct a sample system for verification of the computerized MC/DG concept and develop examples for use in the validation of this sample system

(4) Document the design of the validated conceptual MC/DG system and develop an implementation plan for a full-scale system.

There are several benefits of a computerized MC/DG system:

- Data may be easily kept current by frequent data-base updates (more economically viable than to frequently publish updated hardcopy guides).
- Designer productivity may be increased because:
  - Data required may be easily and rapidly
    - (a) Accessed in a variety of ways via computerized indexing and retrieval
    - (b) Displayed (tabulated, normalized with algorithm, and plotted) in a variety of flexible, user-chosen formats
    - (c) Utilized in automated trade-off analyses
  - By combining the MC/DG with a computer-aided design (CAD) system, many of the tools and data necessary for the designer's tasks are in one location, rather than scattered throughout in various handbooks
- A design history can be accurately maintained by the computerized system for later reference, for example, for use by less experienced designers.

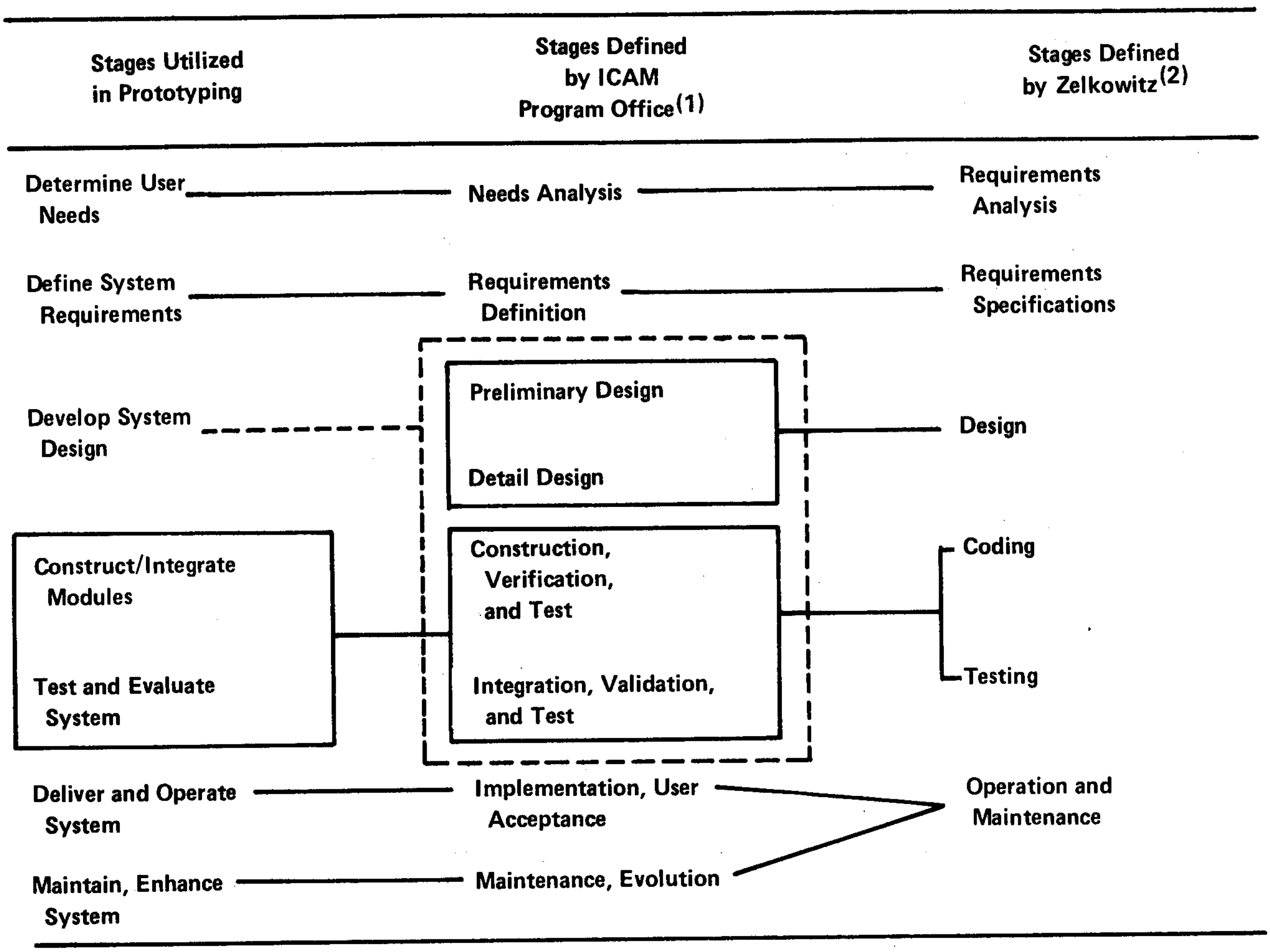
SECTION IV

COMPUTERIZED MC/DG SYSTEM DESIGN FOR CONCEPT VALIDATION

1. THE SOFTWARE DEVELOPMENT PROCESS

To aid in the planning and performance of a software development project, many software specialists have identified discrete software development life-cycle stages. To assist the reader in understanding the particular definition of stages utilized in this description, a comparison of definitions by several authors is shown in Figure 1. In particular, the stages referenced in this report include the following substages:

- Needs Analysis
  - Identify user group
  - Determine user needs
- Requirements Definition
  - Define system inputs/outputs
  - Define data elements and identify sources of data
  - Define system functions (processes to convert inputs to outputs)
  - Define system constraints
- Preliminary and Detail Design
  - Establish levels and types of service
  - Define system functions
  - Define module functions and interfaces
  - Specify system operations (procedures, rules)
  - Document the system design
- Construction, Verification, and Test
  - Construct the modules
  - Verify by testing individual module
  - Specify system test procedures
  - Test major subsystems
- Integration, Validation, and Test
  - Integrate the modules
  - Validate system
  - Test integrated system
  - Correct problems identified



(1) Presented by Mr. Richard Mayer and Mr. David Judson, AFML Integrated Computer-Aided Manufacturing Program, at the NBS Software Requirements Workshop, National Bureau of Standards, Washington, D.C., March 29, 1978.

(2) "Perspectives on Software Engineering", Marvin V. Zelkowitz, ACM Computing Surveys, 10, 2 (June, 1978), 197-216.

FIGURE 1. SOFTWARE DEVELOPMENT LIFE-CYCLE STAGES

- Implementation, User Acceptance
  - Implement system
  - Perform acceptance test of installed system
  - Operate system
  - Perform system monitoring to identify limitations, bottlenecks, errors
- Maintenance, Evolution
  - Correct errors
  - Redesign and implement new modules to correct bottlenecks
  - Evolve new module to correct limitations.

The software development stages discussed above apply to a general software system life cycle. The particular approach utilized for the concept validation system are illustrated in Figure 2. The basic steps performed were:

- Identify System Users. The primary users are the aerospace designers; the secondary user is the person or group who will provide system support services.
- Interview Primary System Users. During the performance of the two MC/DG contracts, aerospace designers were interviewed to determine the characteristics and criteria of the user needs and attitudes toward computerized job aids.
- Develop User Needs Questionnaire and Analyze Responses. Responses were obtained from aerospace company personnel representing a range of age, design experience, and computer usage experience. The results confirm that the MC/DG is needed and would be used in both hardcopy and computerized forms.
- Determine User Needs. Utilizing previous system design experience and the questionnaire results, the user needs were determined for both the designer and systems support services.
- Determine System Requirements. The characteristics and criteria of the user needs were analyzed to determine system requirements.

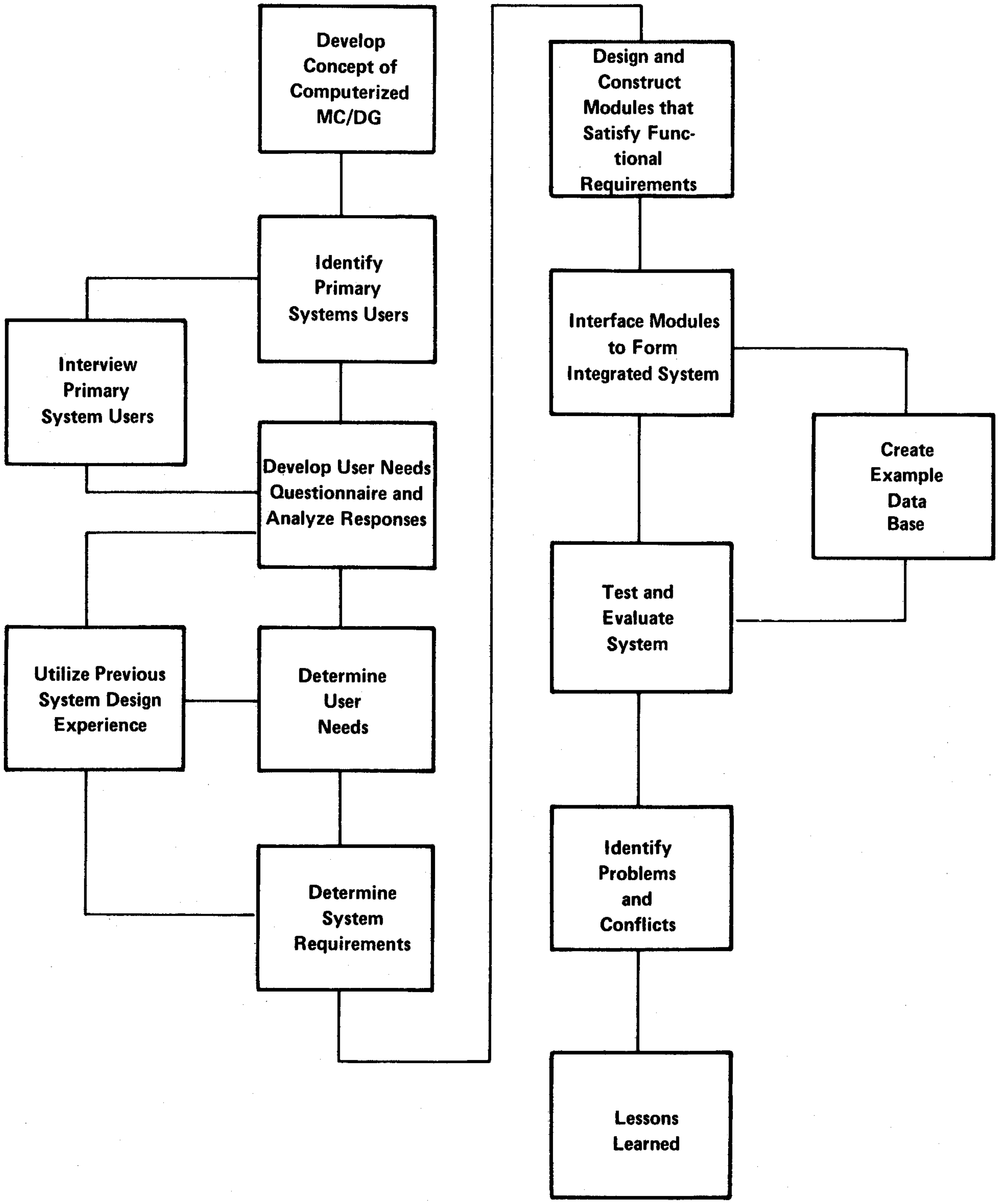


FIGURE 2. COMPUTERIZED MC/DG CONCEPT VALIDATION DESIGN PROCESS

- Identify Modules Which Provide for the Functional Requirements. Available modules were identified that met functional requirements for concept validation.
- Interface Modules to Form Integrated System. The modules were integrated to form a system that served to validate the computerized MC/DG concept.
- Create an Example Data Base. Utilizing the data management module and the man-hour data provided by the aerospace company team members, a sample data base was created.
- Test and Evaluate the System. The concept validation system has been tested and evaluated. As a result, development of a full-scale system is recommended.
- Critique the Example System. The critique mechanism established for the validation of the conceptual MC/DG system includes the use of this report.

2. USER NEEDS

Based upon the definition of the user group (primarily, aerospace designers, and secondly, the support system services) and also interviews with, and survey responses from, potential users, user needs were categorized as follows:

- Aerospace designer needs. The requirement of a systematic and procedurally user oriented system that collects, organizes, retrieves, analyzes, and displays data; the need to identify cost drivers and their effect, to obtain and classify extrinsic and intrinsic cost-estimating and assembly cost data; and the need to perform trade-off analyses.
- Support system services needs. The requirement to organize data, process input data, maintain the base of data and data display formats, and implement, utilizing a user oriented set of hardware and software.

a. Designer Needs

A representation of the aerospace designer needs is illustrated in Figure 3. The higher levels are as follows:

- Learn to Use the System. The most immediate need for use of any new system is training. Instructional media needed are a written user guide, classroom reinforcement of concepts, and on-line (computer-aided) tutorials. The computer-aided tutorials should be packaged in the high-level (macro) procedural language required to support user needs.
- Selection of Parts/Subassemblies (Including Cost/Weight Trade Studies for Alternative Designs. These involve the need to:
  - Retrieve Cost-Driver Effects (CDE) and Cost-Estimating Data (CED). To accomplish retrievals, the user needs to simplistically initiate the use of standard macro procedures for common retrievals and to perform non-standard index term, numeric range term, and sequential search. Also, when more than one retrieval is performed, either searching or retrieval set Boolean operations must be performed to isolate the unique set of data desired.
  - Display Data in Tables and Graphs. To display data, the user needs to simplistically initiate the use of standard macro procedures for common tables and graphs. For specialized tables, the user needs the ability to conveniently and easily specify the composition of tables; for specialized graphs and charts, the user also needs a convenient and simple method to specify the composition of graphs and charts. Display and analysis (trade study) results, as well as retrieved data, are needed.
  - Perform Analysis (Trade Studies). To perform analyses, the user needs a wide variety of simplistically invoked standard macro procedures for common trade studies. For specialized analyses, external program modules will be needed. For specialized analyses, the user also needs

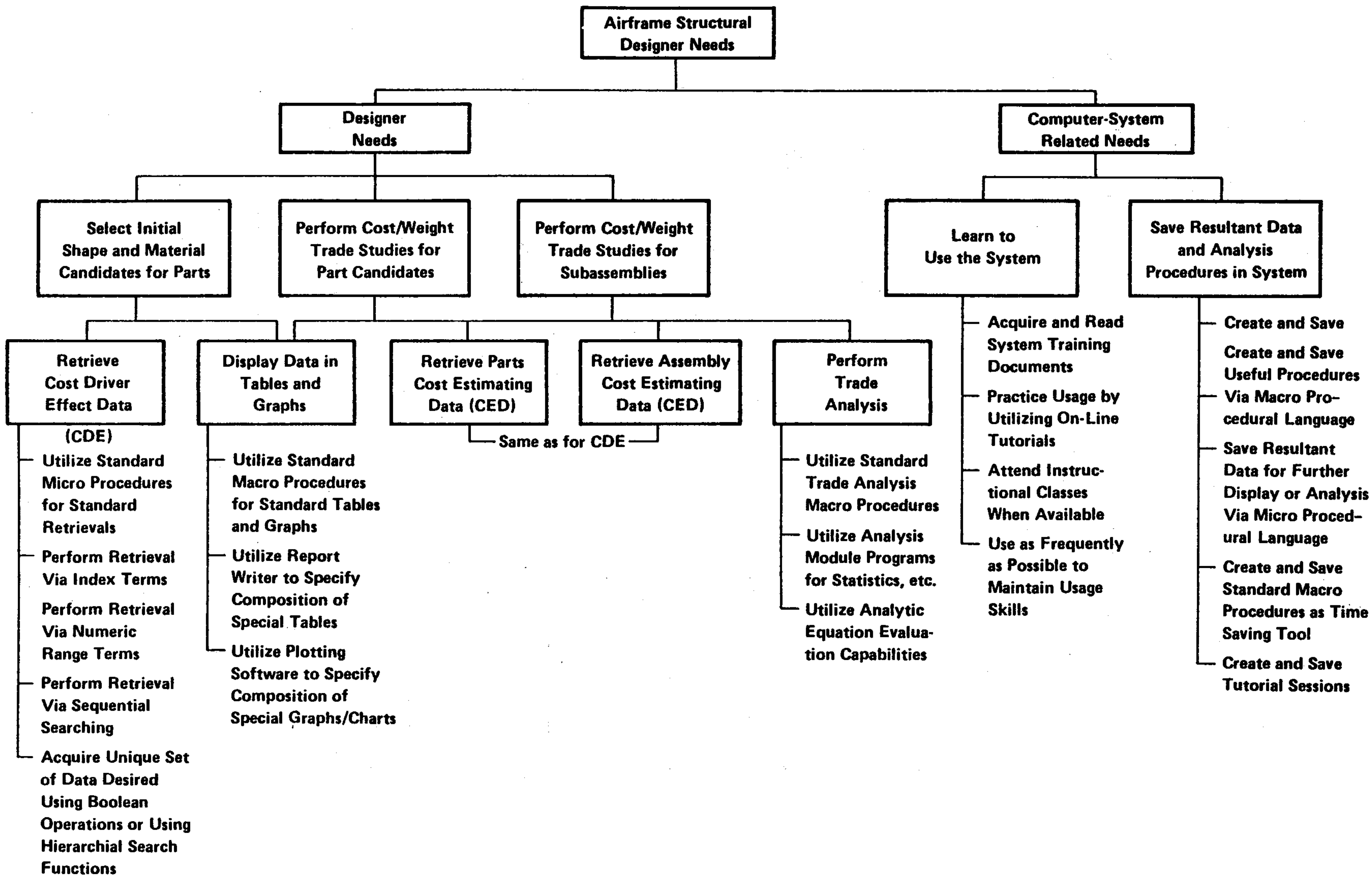


FIGURE 3. DETERMINATION OF AIRFRAME STRUCTURAL DESIGNER NEEDS

to evaluate analytic expressions involving retrieved (and analyzed) data. The ability to save the analysis results for further analysis is needed.

- Save Procedures and Data Analysis Results. The user needs the ability to simplistically create and save unique procedures. The procedures may be for specialized retrievals, displays, or analyses. Also, the user needs the ability to save the results of analyses in a temporary user-assigned storage. This capability was utilized during this contract in storing evaluated man-hour data in a data base.

b. Support System Services

An illustration of support system services needs is shown in Figure 4. The higher levels are as follows:

- Organize Data. The data administrator needs the ability to define the organization of data elements, records, and files. Also, the privacy (access limits) of data needs to be defined. Redefinition of data elements, records, and files is needed so that the system can respond to the changing needs of the designer.
- Process Input Data. The data administrator needs the ability to validate, evaluate (analyze), and format input data for insertion in the data base. An audit trail of input data also needs to be maintained for backup/restoration of the data files in the event of damage to the data.
- Maintain the Base of Data. The data administrator needs the ability to create, edit, add, remove, and restructure data files. Also, a periodic review (monitoring) of the data validity, utility, and organization is needed. The data administrator may optionally delegate responsibility to users for some portions of the data.
- Maintain Data Display/Formats. The data administrator needs the ability to create, edit, add, and remove the description (formats and data elements) of tables, plots,

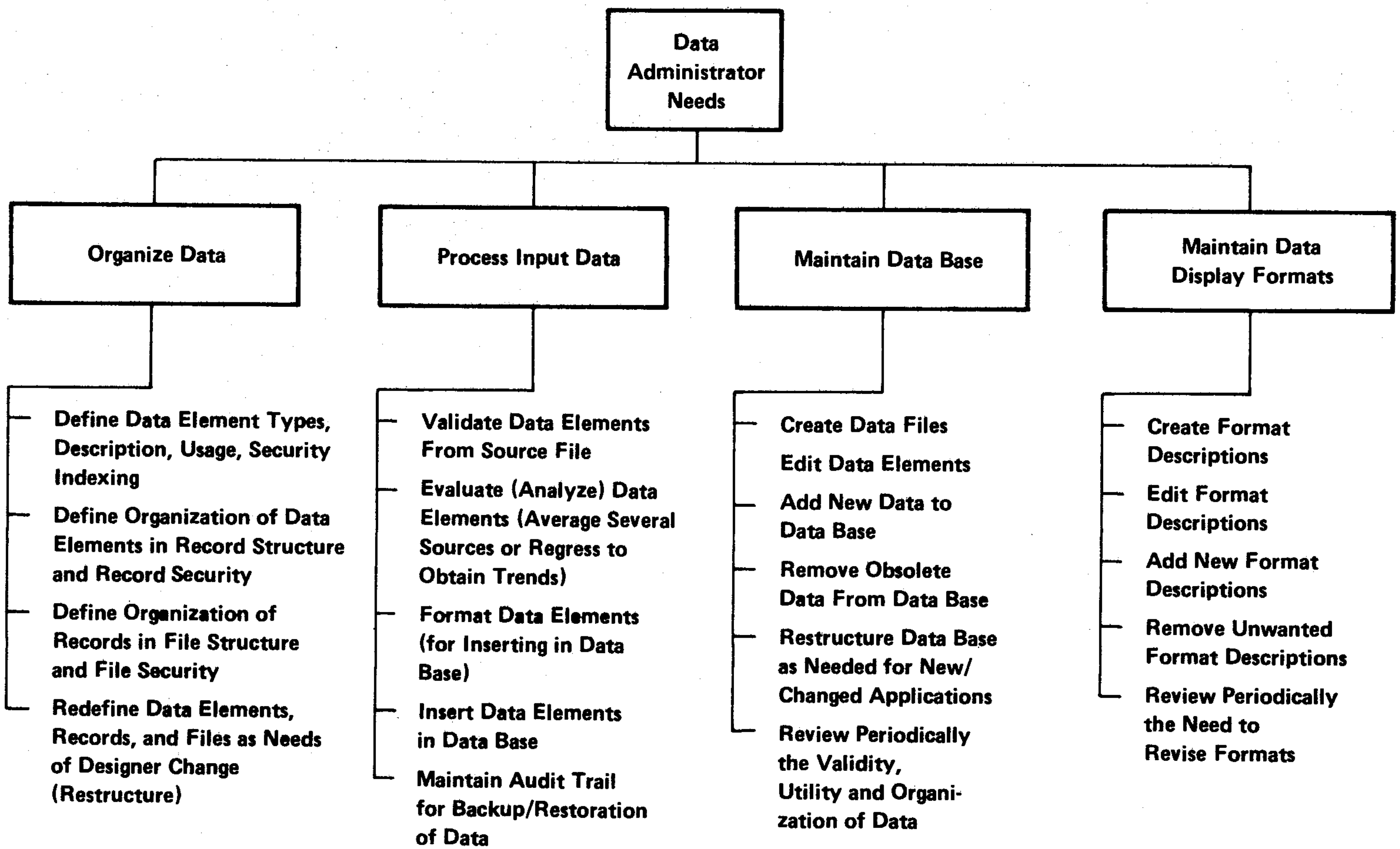


FIGURE 4. DETERMINATION OF DATA ADMINISTRATOR NEEDS

and bar charts. The data administrator should be charged with the maintenance of commonly-used display formats while the user would maintain personalized formats.

3. IMPLICIT NEEDS

Implicit in the concept of an integrated system of independent modules is the need for an integrated user language processing function and control function. That is, the user enters requests (commands) which are to be interpreted as control directives for a particular system module. A simplified logic diagram of request processing for the integrated system is shown in Figure 5. According to the logic depicted, a request may be a basic command for a function to be performed by a module or the request may be a macro procedural language statement. The macro statement is decomposed into statements by the macro processor. By these simple language rules, a macro statement may decompose into basic commands and further macro statements. The macro procedural language may be used for saving retrievals, displays, and trade-off analyses. By collecting macros for simple procedures in a library, higher level macros for tutorials or complex analyses may be developed.

It is desirable that the request processing performed by the user language and System Control Module be capable of informing the user, upon interrogation, about the current system state and which commands are permissible. By system state we here mean identification of module, submodule, program, subroutine, etc.; that is, the state uniquely identifies the software entity being executed. Identification of system state should be accompanied by a very brief functional capability description for the commands permissible at that system state. Detailed state description and command syntax description should be performed by the System Training Aids and User Instruction Module.

4. SYSTEM REQUIREMENTS

From the determination of user needs, the system requirements were defined by performance of the following steps:

- The system inputs/outputs required were specified as a part of other tasks of the program and are described later. The computerized outputs for the concept

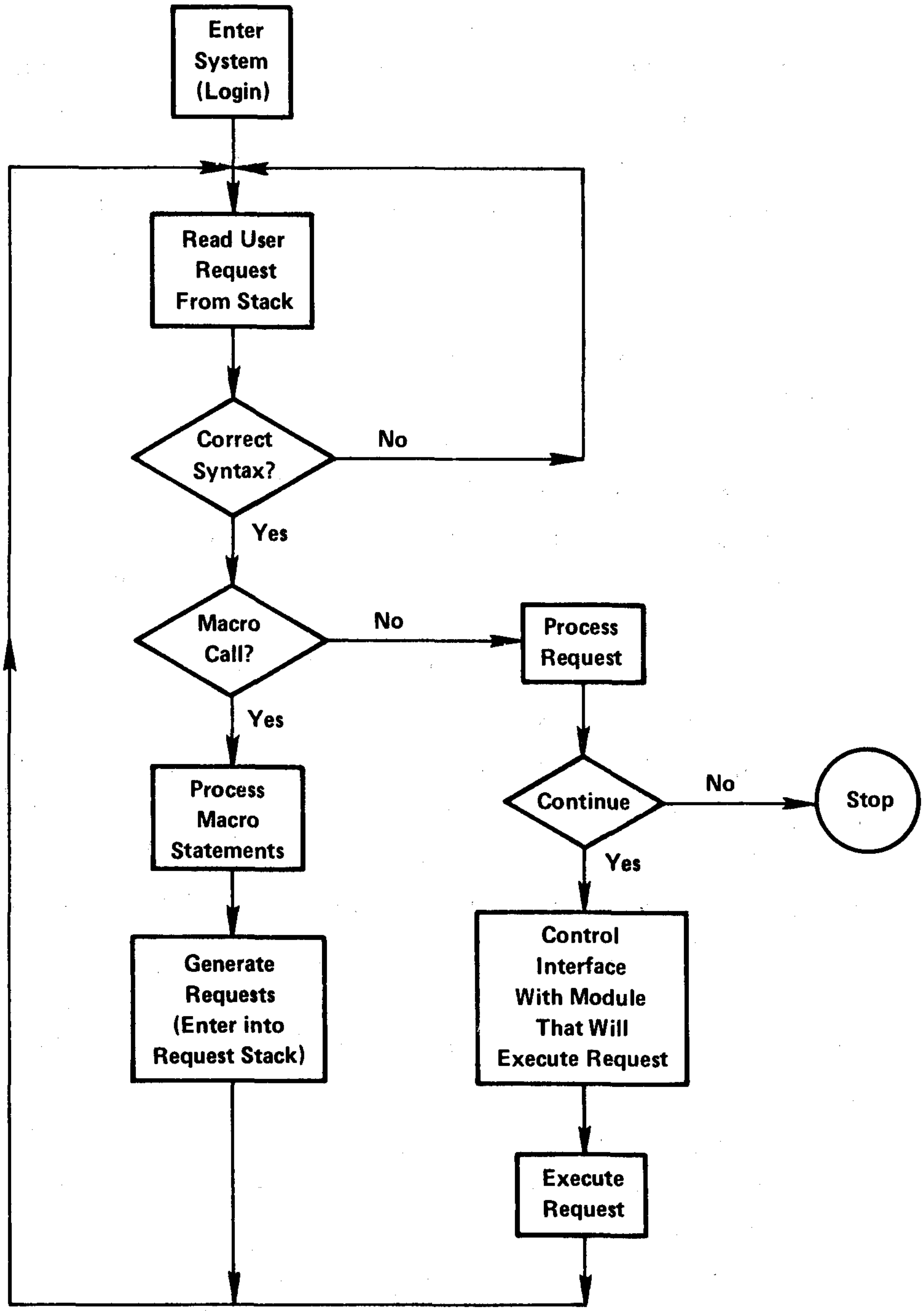


FIGURE 5. MC/DG USER LANGUAGE AND SYSTEM CONTROL FUNCTIONAL REQUIREMENTS

validation study are, in some cases, simplified versions of the hardcopy display formats found in this report. The more detailed display formats would be developed as a part of the full-scale system development. Sample output formats are illustrated in the report section on "Samples of Usage".

- The data elements and sources of data for the MC/DG were determined as a part of other tasks of the program and are also discussed elsewhere in the report. The specific organization of data elements for the demonstration data base for concept validation are discussed in Sections V-2 and V-6, Pages 40 and 52.
- The system functions for the computerized MC/DG were determined as a specific part of the computerization task. The system functions are discussed in the subsequent sections of this chapter.
- The system constraints were defined in parallel with the definition of appropriate data elements and system functions. The general constraints may be described in terms of functions not performed and user audiences not served in a primary way. The MC/DG is primarily aimed at designers, not manufacturing personnel. Also, the MC/DG is intended as a guide, not as a cost-estimating manual.

A consideration of system requirements, categorized by system function, is illustrated in Figure 6. The higher levels of functional requirements are as follows:

- System Task Control. The primary system control requirements are for user language processing and system/module interface control.
- Data Management. The primary data management requirements are for data-base processing (data retrieval and maintenance) and for data administration (data organization and control).
- Data Display and Analysis. The primary data display and analysis requirements are for data tabulation, plotting, arithmetic expression evaluation, and trade-off analyses.

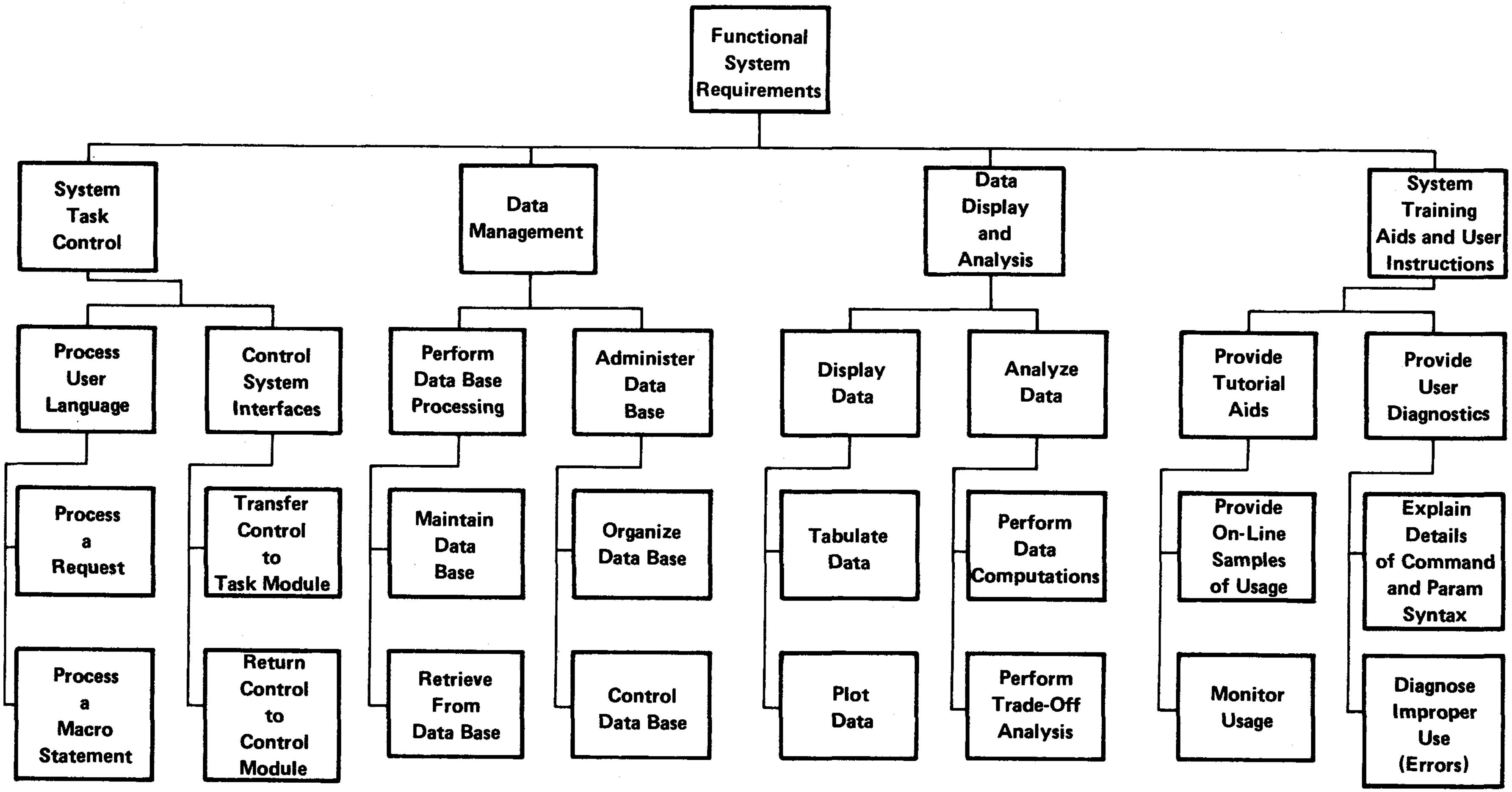


FIGURE 6. FUNCTIONAL SYSTEM REQUIREMENTS FOR A COMPUTERIZED MC/DG

- System Training Aids and User Instructions. The primary requirements for training aids and user instructions are for on-line samples of usage, on-line monitoring of usage, explanations of detailed command/parameter syntax, and diagnostic (error) messages for improper requests.

The following sections elaborate on the four categories of functional system requirements for the computerized MC/DG. The functional requirements presented are those that should be resident in a computerized MC/DG system. The support requirements for operating system functions, language compilers, hardware device access (physical addressing of storage), and tele-processing are best to be considered peripheral to the computerized MC/DG.

a. System Task Control Requirements

The functional requirements for system task control are illustrated in Figure 7. These commands relative to modifying the system will not be available to the average user. Persons responsible for system support, or users more knowledgeable in the use of the computerized system, may need to operate on this level of detail for system modifications and updates, or to create special purpose uses. The functional requirements are categorized as follows:

- Process User Language
  - Process Request (System Command)
    - (a) Read from request stack. A stack (ordered list) of requests (commands) should be maintained for system control. Requests are placed in the stack by the macro procedural language (discussed later) or by the system when soliciting a request from the user.
    - (b) Interpret request. A request, consisting of a command and associated parameters or a macro statement, may have complex syntax. To interpret the request, an interpreter, consisting of a symbol recognizer and syntax analyzer, are required. It would be desirable if the allowed symbols and syntax were contained in tables so that a common interpreter routine could be

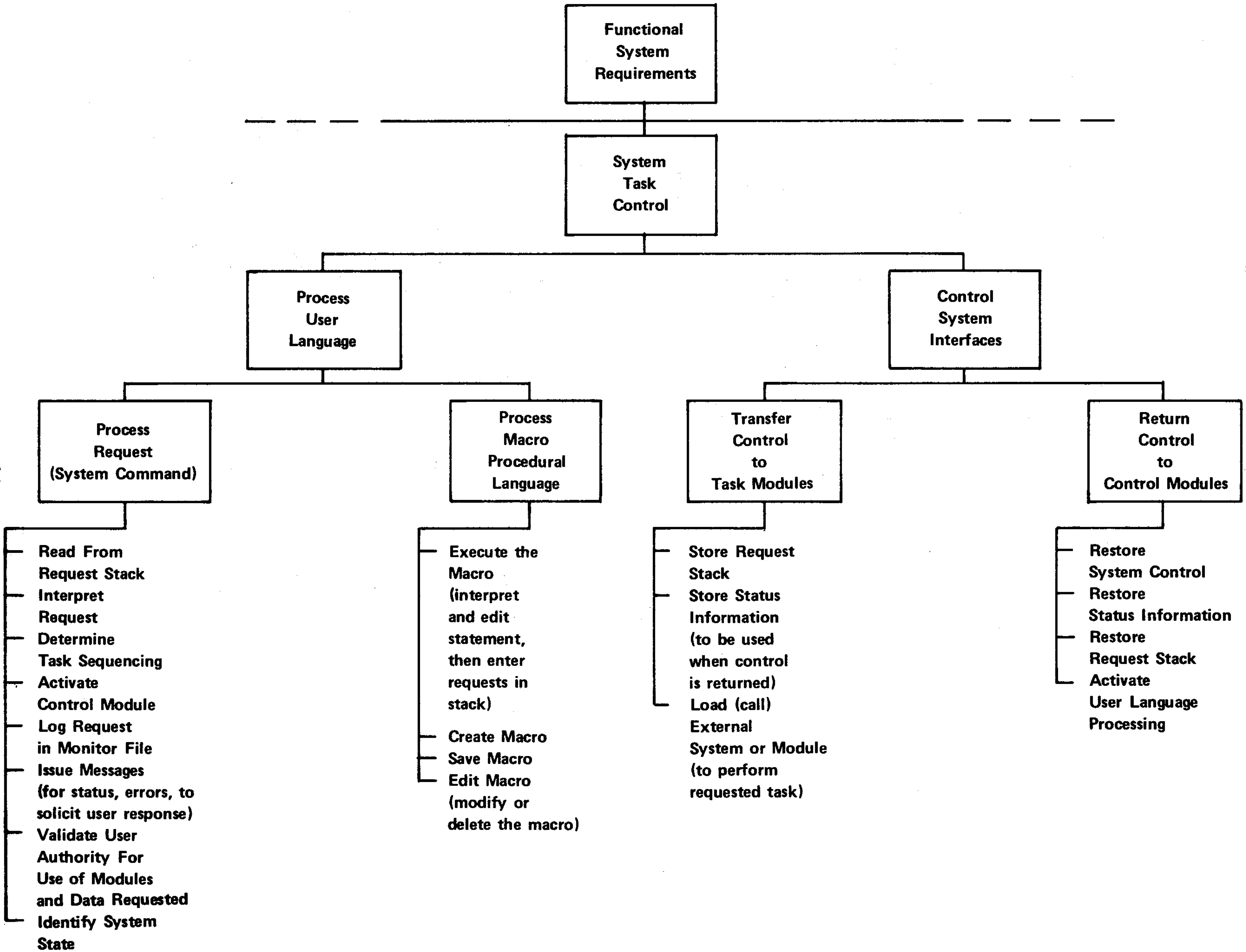


FIGURE 7. SYSTEM TASK CONTROL FUNCTIONAL REQUIREMENTS

used at the top (control) level and at the module level; each module could use its own symbol and syntax tables. The output of the interpreter is a task operations/operands table.

(c) Determine task sequencing. From the task operations/operands table, tasks to be performed by modules are sequenced in an optimal way. This sequencing may be performed as a post-processing step of the interpreter.

(d) Activate control module. Once task sequencing has been established, control should be transferred to the module to perform the task. Upon task completion, control should be returned to the system task control module for execution of the next task.

(e) Log request. To enable system errors, bottlenecks, and operational limitations to be diagnosed, a log of all system requests should be automatically maintained. This log could provide basic data for the design of system enhancements as well as correction of problems.

(f) Issue messages. In conjunction with the other system task control functions, a message handler should provide error messages, explanations, and solicitations for user input.

(g) Validate user authority for module/data usage. At the top level, control of module access and data access must be provided in accordance with the security protocol determined by the organization operating the system. Some modules may contain proprietary methods which would in turn require proprietary data. Entry to these modules by unauthorized users should be a top-level control function. Lower-level controls, within modules and records, may be required in addition to top-level control.

(h) Identify system state. In conjunction with issuing messages, (f) above, it is desirable that the system

be able to identify the current system state, permissible commands, and associated functional capabilities in a brief manner. The mode of display of system state information should be determined as a part of the system design stage.

- Process Macro Procedural Language
  - (a) Execute the macro procedural statement. As a part of processing the user language, a macro procedural language processor is needed. For macro execution, the processor would insert requests in the processing stack.
  - (b) Create macro. The macro processor must support a macro creation function. The macro creation mode may be a part of the macro edit mode (described later).
  - (c) Save macro. The source code of the macro should be saved permanently on a magnetic storage media, such as disc, so that rapid access and usage are possible.
  - (d) Edit macro. Maintenance of the macro source code should be accomplished via an edit function of the macro language processor. Maintenance functions such as adding, deleting, and modifying lines of source code should be done using syntax and techniques commonly used today in vendor-supplied editors.

• <u>Control System Interfaces</u>

- Transfer Control to Task Modules
  - (a) Store request stack. The technique suggested for control of modules is to have only one module loaded in core memory at a time. To accomplish this, information needed for control must be saved when the system task control module is swapped out and the module for task performance is swapped into core. The request stack is one such piece of control information that must be saved when control is transferred from the system task control module and must be restored when control is transferred back to the system task control module.

(b) Store status information. System and module status information that must be retained throughout a user session with the computerized MC/DG must be stored and restored each time control is transferred from one module to another.

(c) Load (call) external system or module. The actual loading of systems/modules into the computer is most efficiently performed by the host operating system loader. The function to be performed by the computerized MC/DG system is to issue the proper operating system load (call) directive. In the memory management technique suggested in this system requirement description, modules are independent, stand-alone programs to be loaded into computer core. In turn, each module may utilize memory management techniques such as overlays or segmentation to the extent supported by the language compilers and host operating system.

- Return Control to System Task Control Module

  (a) Restore system control. When execution of a task module is complete, the module should issue a load (call) directive to bring the system task control module into core memory.

  (b) Restore status information. When the system task control module is restored, it should immediately restore the status of information.

  (c) Restore request stack. When the status information has been restored, the request stack must be restored.

  (d) Activate user language processing. When control has been restored to the system task control module and the request stack is empty, the user language processing should be activated, i.e., the user will be asked for a request.

b. Data Management Requirements

The functional requirements for data management, illustrated in Figure 8, can be categorized as follows:

- <u>Data Base Processing</u>
  - Maintain Data Base
    - (a) Store data. One of the basic requirements of data management is the storage of data on a random-access device so that rapid retrieval of data may be accomplished. The computerized MC/DG system should utilize vendor-supplied storage addressing techniques so that the MC/DG system can be as transportable as possible.
    - (b) Create and update data. One of the common data management functions is creation and updating of data in a data base. Creation may entail establishment of new records or entirely new files. Updating consists of deleting existing records or modifying existing records by adding, deleting, or changing data elements within the record.
    - (c) Edit data. The notion of data editing is used in this requirements description as the process of generating transactions for creating and updating data.
    - (d) Restructure data. When a change in objectives or of functions in a system takes place, a restructuring of the data organization is often required. The nature of the restructuring, supported by the system, is limited by the flexibility of the data-base management system (module) procured or developed. At a minimum, the restructuring of data elements within records should be supported by the computerized MC/DG system. Other types of data restructuring will be dependent upon the file organization chosen for the data-base design.
    - (e) Index data. The type of indexing utilized will depend upon the data-base management system procured or developed. It is necessary that flexible, efficient

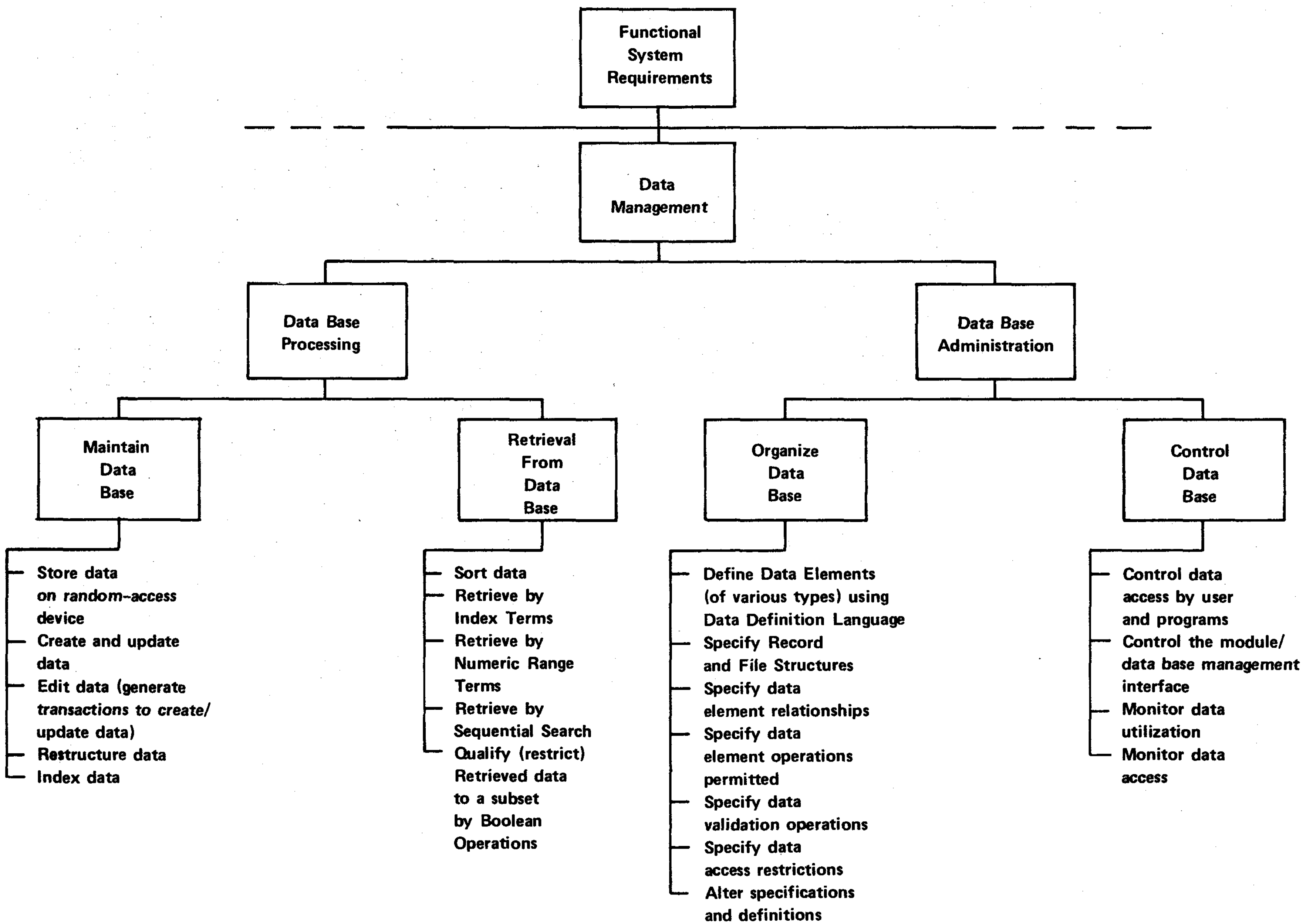


FIGURE 8. DATA MANAGEMENT FUNCTIONAL REQUIREMENTS

indexing be utilized as an alternative to sequentially searching the entire data base to accomplish data retrieval.

- Retrieve from Data Base
  (a) Sort data. Sorting of data is categorized with data retrieval functions because sorting is often required since data retrieval generates data in undesired order. Sorting by multiple sort keys should be a standard MC/DG function. Sort keys should be numeric or textual values from the retrieved data elements as dictated by user requests.
  (b) Retrieve by index terms. MC/DG data should be retrievable by using index terms. The syntax of the retrieval request and the organization of the index will be dependent upon the particular data management system procured or developed. An example of retrieval by index terms would be to select all data for aluminum sheet metal parts.
  (c) Retrieve by numeric range terms. MC/DG data should be retrievable by numeric range terms. As for index terms, the syntax of the request and the organization of the numeric range file will be dependent upon the particular data management system procured or developed. An example of retrieval by numeric range would be to select all data for part lengths between 6 and 12 feet.
  (d) Retrieve by sequential search. MC/DG data should be retrievable by sequentially searching the data base using criteria similar to those used for index terms or numeric range terms. The essential difference is that sequential searching must apply to search criteria to each record to obtain a subset that meets the criteria. This technique should be available for cases where adequate index terms or range terms were not generated. It would be highly desirable for range searching to be performed on a previously retrieved subset of the data base; this would reduce the operational cost of sequential searching.

(e) Qualify (restrict) retrieved data to a subset using Boolean operations. The ability to qualify a retrieval set using Boolean operations is to select all aluminum or steel parts and those between 6 and 12 feet (here the Boolean operators are: or, and).

- Data Administration
  - Organize Data Base

    (a) Define data elements. The definition of data elements and organizational attributes of the data base is best accomplished using a Data Definition Language (DDL). The advantage of maintaining data base dependent information, such as data element descriptions, in a separate file is that the software can be data base independent and the data definition file can be easily modified (the data definition file is usually very small compared to software and data files).

    (b) Specify record and file structures. Record and file structures should be specified using the DDL. The types of file and record structures available must depend upon the particular data-base management system procured or developed.

    (c) Specify data element relationships. Data element relationships should be specified using the DDL. The types of relationships that can be specified will depend upon the particular data base management system procured or developed.

    (d) Specify data element operations. The data element operations permitted will depend to a great extent upon the functions to be performed on the data. At a minimum, numeric and textual data elements should be permitted in tabulations and plots; numeric data should be usable in numeric expression evaluation. The operations permitted should be defined as a part of the DDL but may be implicit in the design of the display and analysis modules.

(e) Specify data validation operations. The DDL should be used to specify data validation procedures. Examples of data validation criteria are: data value must be numeric between 0 and 100; data value must be a date later than 1950 and earlier than 1980; data element must have one of the following values: UNCLASSIFIED, CONFIDENTIAL, or SECRET.

(f) Specify data access restrictions. Data access control protocol should be specified using the DDL. The possible types of access control are dependent upon the data management system procured or developed. Examples of access restrictions are: user must belong to a particular organizational entity; user code must be greater than a record access code for access; user may have access to a particular module but may have access to data for only some of the functions of the module (access restrictions on both data and modules).

(g) Alter specifications and definitions. Since changes to data bases are inevitable, it is necessary to be able to alter specifications of data elements, data organization, data relationships, data validation procedures, and access restrictions. The requirement for altering data specifications is directly related to the data maintenance function of data restructuring.

- Control the Data Base

(a) Control data access. The restriction of some data to authorized users and/or modules is required. An algorithm for control must be developed for each data base so that user, data, and module access codes will determine accessibility. The types of algorithms possible will be dependent upon the data base management system procured or developed.

(b) Control the module/data base management interface. Control of module interfaces is to a large extent

controlled by the system task control module. However, in the case of data management, it is often wise to further control the interface of the data management module with other modules. It is highly desirable that only the data management module has the authority to create and update data and that data creation/update transactions be accepted from only certain well-controlled module interfaces. It is highly recommended that a very small number of authorized users have access to modules that can change the data in the MC/DG data base in order that the integrity of data be maintained.

(c) Monitor data utilizations. The utilization of data should be frequently monitored to assure that the data elements, organization, and operations are optimal in the changing environment of usage. Identified problems can then be remedied through data restructuring and/or software enhancement.

(d) Monitor data access controls. Although intentional breach of access controls is difficult to discover, measures for monitoring access controls is required. The most likely benefit from monitoring is that unintentional breaches of access controls (due to faulty system or access algorithm design) will be identified.

c. Data Display and Analysis Requirements

The functional requirements for data display and analysis, illustrated in Figure 9, can be categorized as follows:

- <u>Data Display</u>
  - Tabulate Data

    (a) Specify tabulation composition and data content via direct report writer commands. A definite requirement of a system such as the computerized MC/DG is

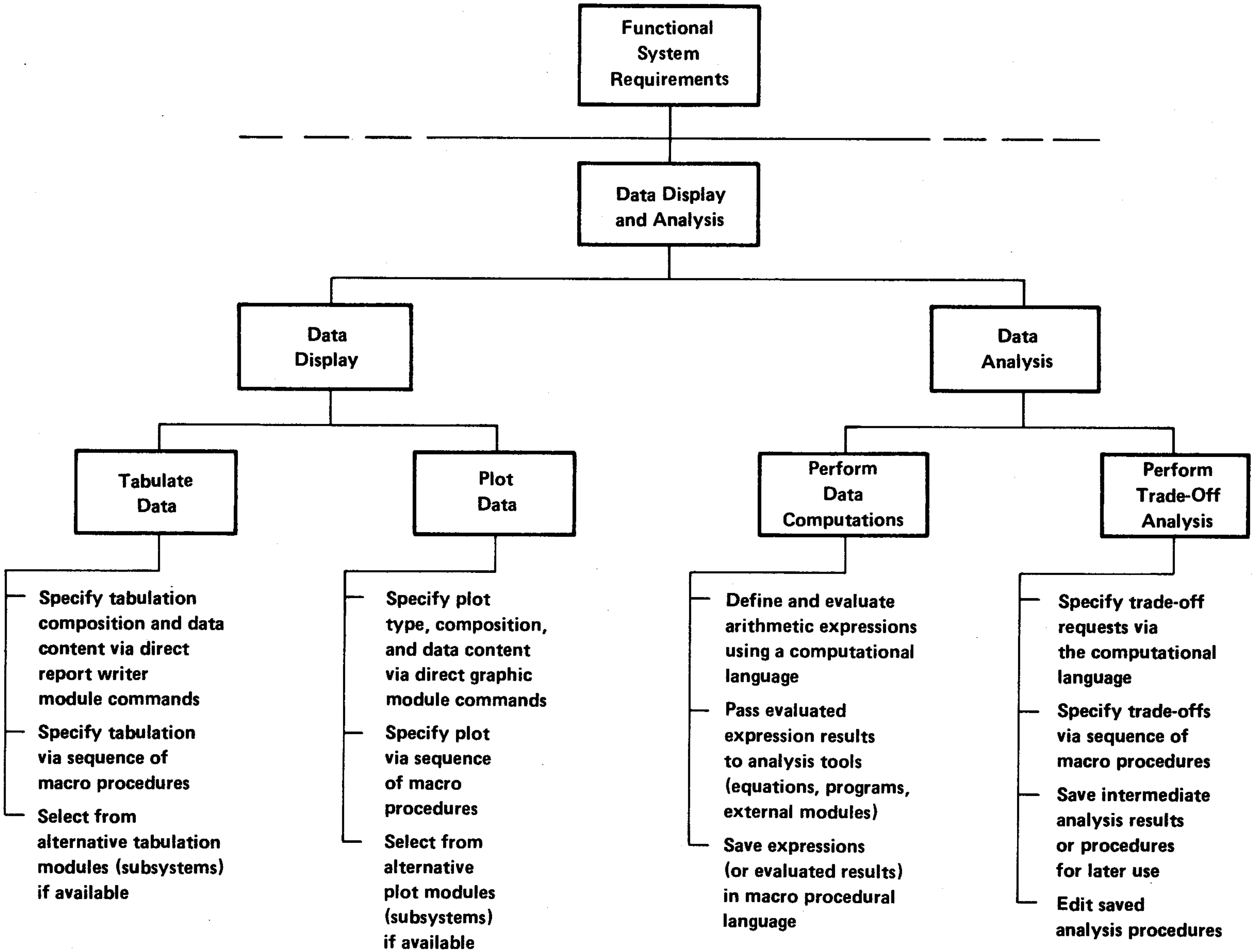


FIGURE 9. DATA DISPLAY AND ANALYSIS FUNCTIONAL REQUIREMENTS

the report writer module. In some systems, certain modules have simplified data display functions in addition to the sophisticated report writer capabilities. The syntax and capabilities of report writers varies widely among presently available software products. However, at a minimum, a matrix style tabulation of data elements with meaningful row, column, and page headings is required. The data element class (integer, floating point, date, multi-line text, etc.) and page position specifications should be easy to use yet flexible.

(b) Specify tabulation via macro procedures. The report writer specifications may be lengthy enough that commonly used tabulations should be saved via the macro procedural language.

(c) Select from alternative tabulation modules (subsystems) if available. Depending upon the capabilities of the data base management system procured or developed, a choice may exist between the data base manager report writer capabilities and the capabilities of the independent report writer module. The options should be preserved so that the most efficient usage of computer and human resources can be achieved.

- Plot Data

  (a) Specify plot type, composition, and data content via direct graphic module commands. The user should be able to easily generate simple plots by specifying the plot type (x-y plot or bar graph), the composition (grids, labels, etc.), and the data content (data elements to be plotted).

  (b) Specify plot via macro procedures. Specifications for complex plots and even for commonly used simple plots should be saved for repeated usage via macro procedures.

(c) Select from alternative plot modules. If alternative plotting modes (such as teletype scatter diagrams, graphic terminal plots, or pen plotter outputs) are available in various modules, the flexibility of the multiple modes should be preserved when the modules are integrated to form the computerized MC/DG. The user should retain the option of selecting from the alternatives.

- Data Analysis
  - Data Computations

(a) Define and evaluate arithmetic expressions using a computational language. A computational language for defining and evaluating arithmetic expressions is required to support trade-off analyses. The language must be flexible enough that the data elements for selected cases may be incorporated in analytical expressions. The arithmetic operations required are: addition, subtraction, multiplication, and division. It is desirable to have special mathematical functions such as logarithms, trigonometric functions, and exponentiation supported by the computational language. The additional ability to qualify or test the expression results via relational and Boolean operators (e.g., AND, OR, NOT) would be desirable.

(b) Pass evaluated arithmetic expression results to analysis tools. It is necessary to pass the evaluated expression results to analysis tools to be used for trade-off results. The analysis tools may be: analytic expressions (that is, use previously evaluated expression results as subexpressions); standard analysis modules within the MC/DG system; external modules supplied by the user.

(c) Save expressions in macro procedures. It is very likely that commonly used trade-off analyses will be

complex. It is, therefore, necessary that the computational expressions and requests for analysis modules be saved in the macro language.

- Perform Trade-offs
  - (a) Specify trade-offs via the computational language. The optimal way of performing trade-off analyses should be via the computational language. If the analysis is very complex, the computational language should support calculation of parameters for analysis modules.
  - (b) Specify trade-offs via macro procedures. The use of macro procedures for trade-off analyses is probably one of the best examples of the need for a macro language. The trade-off analysis may require complex calculations and externally supplied analysis tools; all of the statements to perform the analysis are best packaged in a macro so that testing, refinement, and repeated operational use of the trade-off analysis may be performed.
  - (c) Save intermediate analysis results or procedures. The macro language should be capable of saving one-time usage procedures so that lengthy analyses may be segmented in parts. If, for example, a lengthy analysis had to be continued the next day, the current results and procedures could be saved as a macro. When the analysis is finished, the segments of the analysis could be deleted from the computer using the macro editor.
  - (d) Edit saved analysis procedures. In the process of testing and refining a macro for trade-off analysis, it is necessary to change the macro. The macro editor should be capable of editing the full range of complex analysis macros.

d. System Training Aids and User Instruction Requirements

The functional requirements for system training aids and user instruction, illustrated in Figure 10, can be categorized as follows:

- Tutorial (Teaching) Aids
  - Provide On-Line Samples of Usage
    - (a) Provide many levels of usage samples. The best way to illustrate the usage of system features is via many levels of samples of usage. Some samples should illustrate various usages of commands and parameters; samples of all commands should be illustrated at this level. Higher level samples should incorporate typical sample sessions.
    - (b) Provide samples of usage for many levels of user experience. Samples should be developed that are appropriate to the level of experience of the user; that is, novice and expert user tutorials should be developed.
    - (c) Provide samples of many modes of usage. Some of the samples should concentrate on retrieval/display mode of usage while other samples should concentrate on sophisticated trade-off analyses.
    - (d) Provide samples of combined usage modes. Some complex samples of usage should be provided which challenge the user to propose alternate approaches to a computerized MC/DG session.
  - Monitor Data
    - (a) Collect on-line samples of usage. The sessions of novice and expert users should be monitored as a way of identifying candidate sessions for tutorials.
    - (b) Analyze user modes of usage. The modes of usage of real on-line sessions should be analyzed to identify improper usage, new modes of usage being developed by some expert users, and to identify system limitations. New modes of usage identified should be illustrated in tutorials.

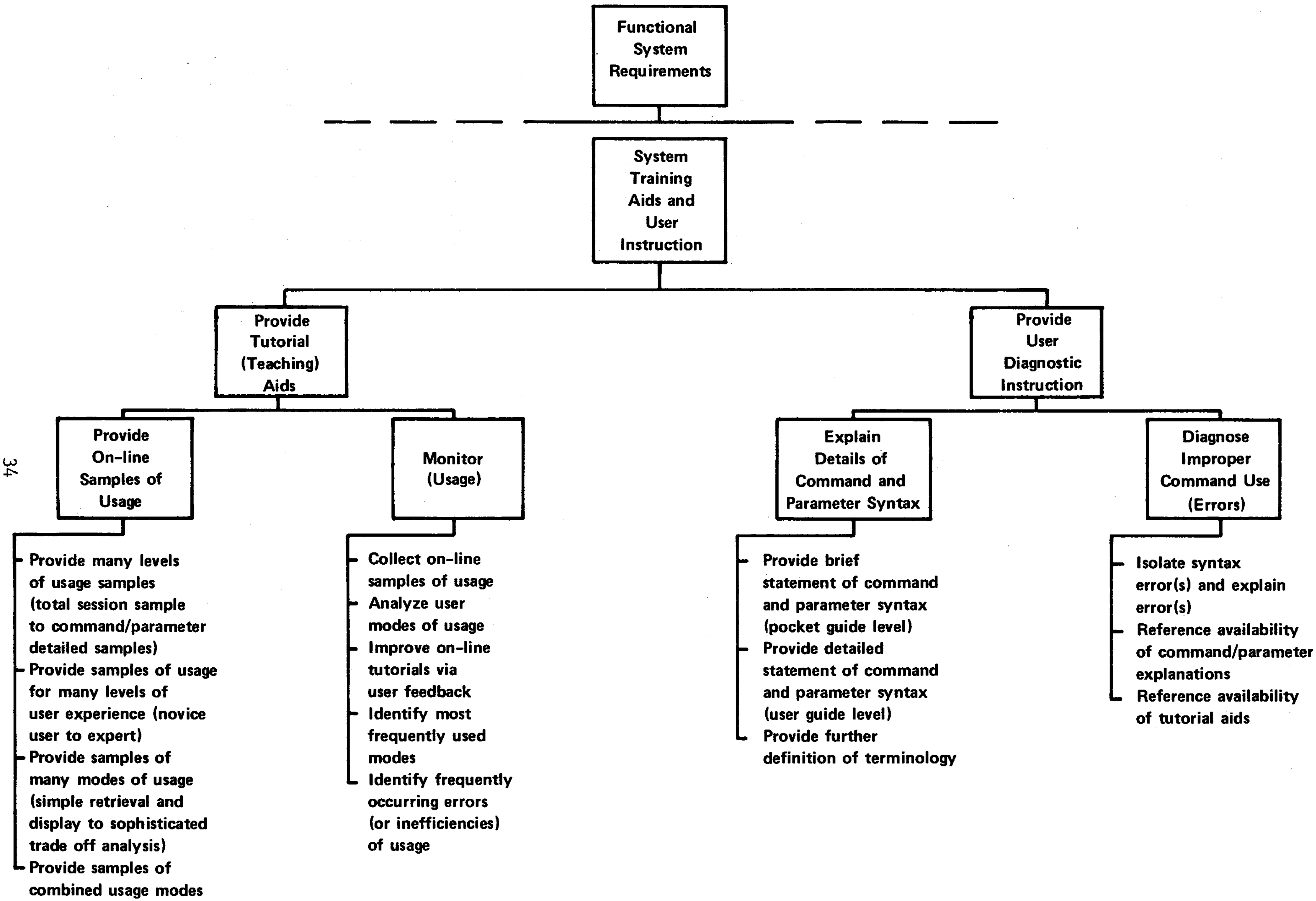


FIGURE 10. SYSTEM TRAINING AIDS AND USER INSTRUCTION FUNCTIONAL REQUIREMENTS

(c) Improve on-line tutorials via user feedback. Shortcomings of initially developed tutorials should be identified and corrected via critiques from users.

(d) Identify most frequently used modes. The most frequently used modes of system usage should be given additional attention for tutorial and macro procedure development/enhancement.

(e) Identify frequently occurring errors. The statistics from monitor data should analyze the most frequent errors or inefficient usage patterns. Once identified, this information should form the base of an enhancement effort to improve training and software design.

- User Diagnostic Instruction
  - Explain Details of Command/Parameter Syntax

    (a) Provide brief statement of command/parameter syntax. For the expert user, the equivalent of "pocket guide" level messages should be provided.

    (b) Provide detailed statement of command/parameter syntax. For the novice user, detailed explanations equivalent to "user guide" narrative should be provided.

    (c) Provide further definition of terminology. Any diagnostic message terminology not understood by user should be further explained via an explain command.

  - Diagnose Improper Command/Parameter Use

    (a) Isolate syntax error(s) and explain error(s). Immediate system diagnostic messages should be issued when syntax errors are identified.

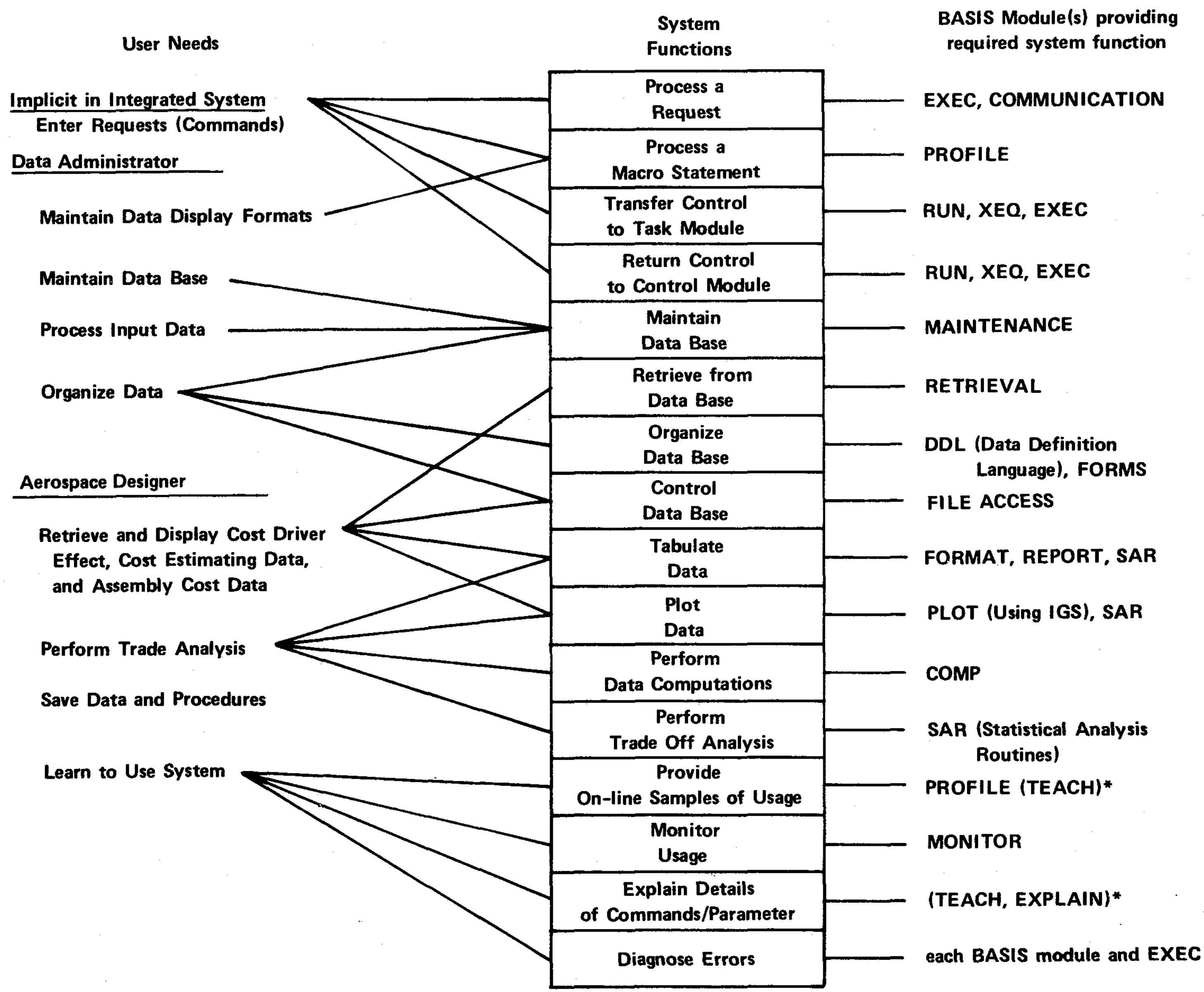
    (b) Reference availability of command/parameter explanations. Identified syntax errors should cause a message to be issued to reference to availability of explanations of command/parameter syntax.

    (c) Reference availability of tutorials. Identified syntax errors should cause a message to be issued to reference the availability of tutorials illustrating the proper use of the command/parameters.

5. THE DEMONSTRATION MC/DG SYSTEM

Based upon the system functional requirements discussed in the sections above, a demonstration computerized MC/DG system was constructed utilizing existing modules of the BASIS (Battelle Automated Search Information System) and the IGS (Integrated Graphics System) software. Figure 11 illustrates the overall correlation of user needs, functional system requirements, and BASIS modules satisfying the functional system requirements.

The primary features of the concept validation system are illustrated in Figure 12.



*To be implemented.

FIGURE 11. CORRELATION OF USER NEEDS, SYSTEM FUNCTIONS, AND BASIS MODULE CAPABILITIES FOR THE CONCEPT VALIDATION SYSTEM

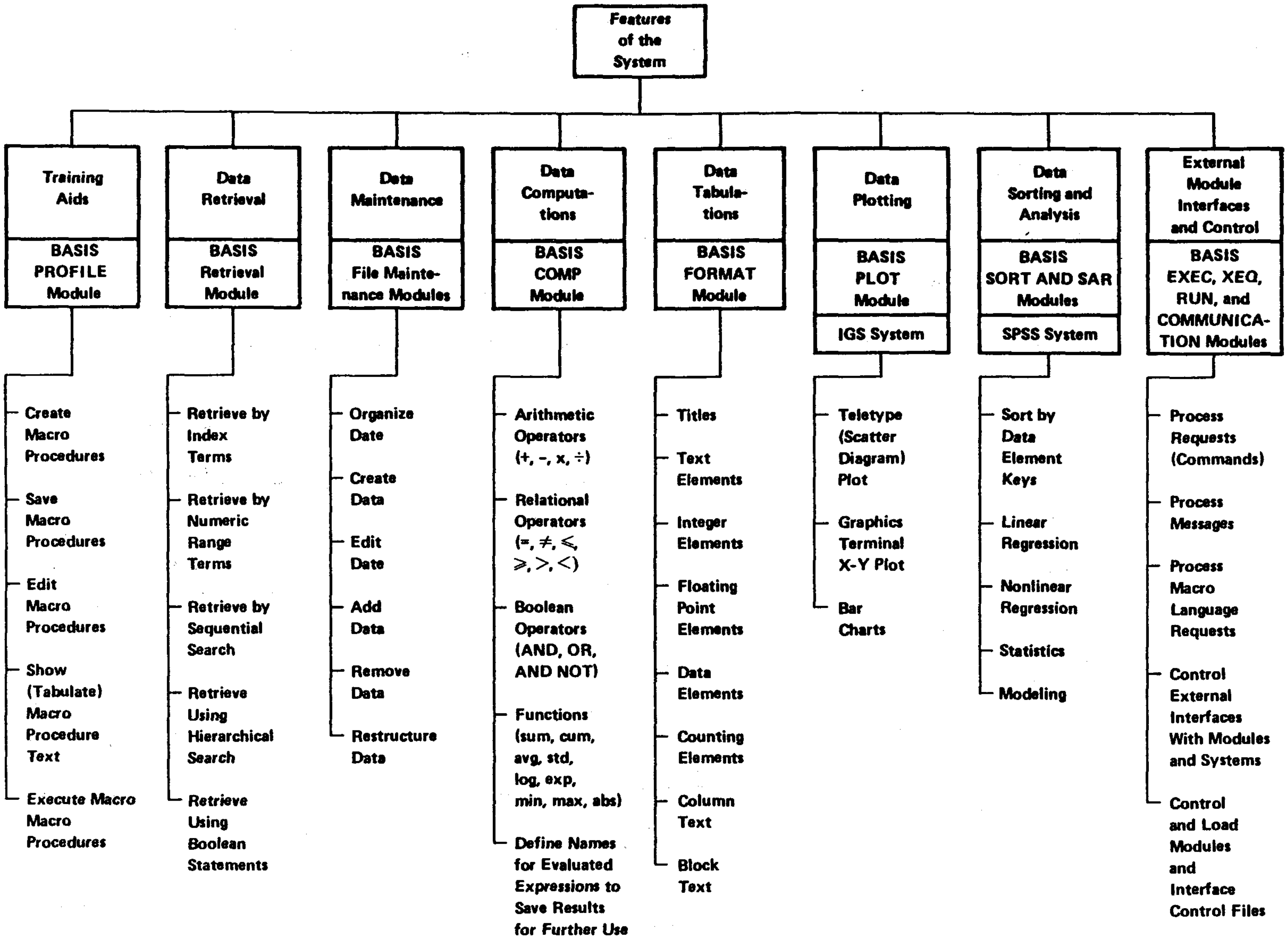


FIGURE 12. FEATURES OF THE COMPUTERIZED MC/DG SYSTEM FOR CONCEPT VALIDATION

SECTION V

THE DEMONSTRATION MC/DG SYSTEM

1. PROCEDURE USED TO DEVELOP MC/DG MANUFACTURING PROCESS MAN-HOUR (COST) DATA

The following procedure was used to develop the manufacturing man-hour data for the development of designer-oriented CDE and CED formats:

- Analyze engineering drawings and determine manufacturing technology category, e.g.,
  - Fabrication
  - Machining
  - Assembly
- Establish operational element sequences, including equipment requirements
- Apply I.E. base standards to each operational element; T1000 used
  - Set-up cycle
  - Run-cycle

  (Applied standards are the sum total of all elements required to carry out each operation plus set-up.)
- List tooling required for each operational element
- Total set-up cycle hours for each operational element
- Amortize total set-up hours over lot size and add to total run-cycle time.

The result of the above is an applied standard for unit part cost (applied standards are productive, hands-on, or direct factory labor hours only).

The data developed by the aerospace company team members were normalized for CDE and CED formats by BCL.

To establish T_u200 cost (from T1000), selected improvement/learning curves are used (no variances included; for example, PF&D, clean-up, rest period, equipment down-time, and supervisory instruction). For individual companies to determine the cost of a discrete part, actual labor rates must be used and material cost must be incorporated.

2. DESCRIPTION OF THE PHASE I DATA BASE

The Demonstration MC/DG Phase I data base will be described as follows:

- Data base structure and storage
- Data base construction procedures
- Data retrieval.

a. Data Base Structure and Storage

The sample data base was constructed and managed by utilizing the Battelle Automated Search and Information System (BASIS). It consists of the following six files, each of which has specific functions (see Figure 13):

- The Communication File contains all the information for each terminal session
- The Head or Source Data File contains all the source data information.
- The Index File contains information necessary for the rapid retrieval of data items and data records.
- The Message File contains system messages and special design dialogue for display at a user terminal.
- The Range File is used in conjunction with the Index File for fast numeric data information retrieval.
- The Table File contains description and location of each of the other data base files, and an array of descriptive tables which the system requires to process the data base.

The Source Data File consists of three logical subfiles:

- The Raw Data Subfile--stores the raw data submitted by the five team member companies. The data records and the data source field are secured, in that, only users with an authorized password may access them.

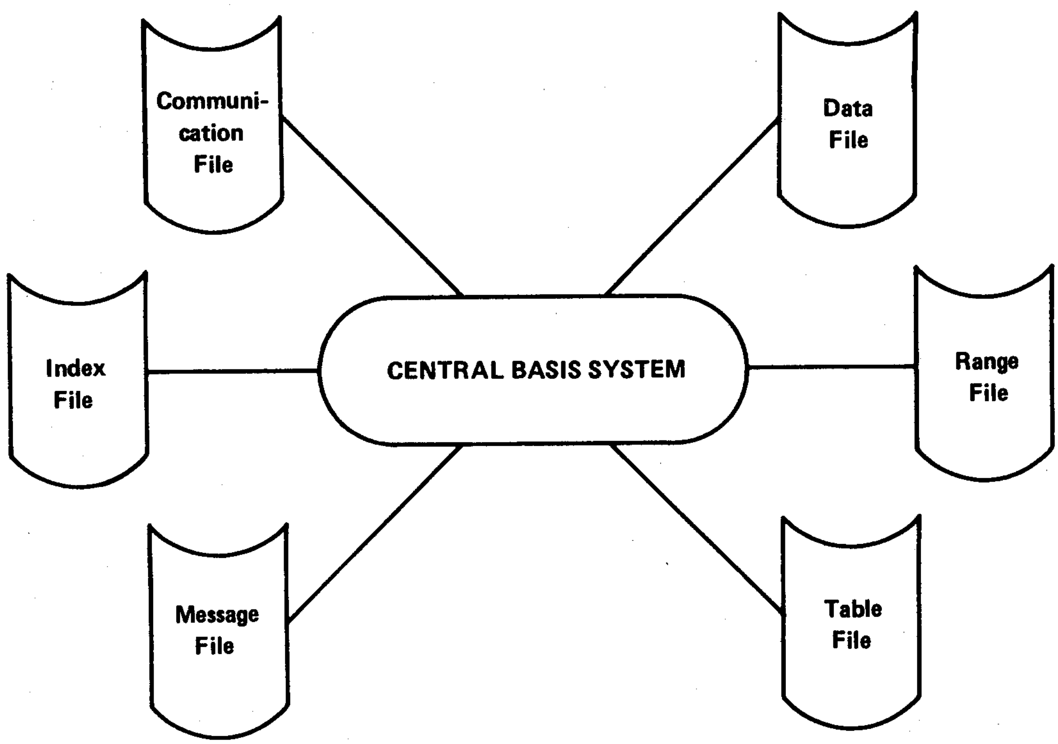


FIGURE 13. THE DEMONSTRATION MC/DG DATA BASE FILES

- The Average Data Subfile--contains the data computed by averaging the sum of individually itemized man-hour costs over the number of non-zero data items. It is this set of data records that was retrieved for demonstration purposes.
- The Statistics Subfile--consists of computed values for statistical analysis of the raw data, e.g., regression analysis coefficients.

Each data record is uniquely identified by an accession number, a 9-digit integer that contains the codes for material type, discrete part code, manufacturing method, production lot size, data source, and part measurements.

The entire data base resides in disk storage for rapid on-line access. Backup copies of the data base files are stored on magnetic tapes.

c. Data Base Construction Procedures

The following subtasks are key to processing data for computer presentations:

- Prepare and review data collection forms
- Process data
- Review processed data
- Categorize data by presentation mode (i.e., tables or plots).

(1) Data Collection Forms

Several draft data collection forms were prepared. Each of the team members conducted tests on the draft data collection forms during the time period between the first and second team meetings. At the second team meeting, data collection formats were agreed upon.

The manufacturing man-hour data for 20 aluminum, steel, and titanium sheet metal discrete parts made by different manufacturing methods were submitted by the five team members on such forms.

An example of a data summary sheet is shown in Figure 14, Page 59. One form summarizing the manufacturing costs for lot sizes of 5, 10, 25, and 50 is shown in Figure 15, Page 60.

(2) Process Data

Upon receipt of data from each team member, BCL engineers screened and processed the manufacturing method man-hours. The screening process entailed manually comparing the costs from each company to determine the cause of data variation, for example, the availability of unique manufacturing facilities or the estimation method employed by an airframe company. Special care was taken to prevent variations in the data since both general and detailed ground rules were established before data collection was initiated.

After careful screening for obvious errors, the data were then coded and keypunched on cards. The data cards were processed by a Summary/Average program which prints out the data summary reports, as shown in Figure 15 (Page 60), for the raw data and the averaged data for further visual verification. This was to ensure the accuracy of the data prior to its entry into the data base Source Data File.

When all data entry keying errors were rectified, the data cards were processed by an Input-Preprocessing program which transformed the coded data items into their proper formats, as shown in Figure 16 (Page 65), acceptable to the Data Base Management Systems.

The Input Processor program then presented the decoded data as transactions ready to be entered into the Source Data File, the Index File, and the Range File by the respective File Managers in the BASIS File Maintenance Module.

(3) Data Retrieval

The data record retrieval specification is described within the Data Definition Language (DDL) program. Table 1 (Page 67) shows the MC/DG Phase I sample data base record elements. Each data element is termed a "field", identified by the data base management system through a unique field number and a field name. It is displayed according to a specified format, with a unique display label; unit labels for display are optional. Within the computer, all data elements are stored as packed alphanumeric character strings of varying length. Fields without values are not stored; for example, data records for the aluminum 2024 beaded panel manufactured with rubber press process require no other

manufacturing complexities, such as joggles, flanges, end or lineal trim, etc., hence, do not have values for fields 28, 29, 32, and 33. Furthermore, leading zeros in numeric fields and trailing or redundant embedded blanks in alphanumeric fields are discarded. Consequently, the data records are of variable length, thus optimizing computer storage space.

Data retrieval criteria are established in the Indexing Definition section of the DDL program. Numeric fields are retrieved via range value search, e.g., entering these requests

(1) NRTCOST LE 3.00

(2) NRTCOST GE 1.00

(3) 1 AND 2

will result in retrieving the set of data records in the data base for those parts with non-recurring tooling cost between 1 and 3 manufacturing man-hours.

Textual fields can be retrieved by free text indexing or term indexing. Free text indexing allows the user to retrieve records containing a common word or a phrase within a specified textual type of data field. For example, field 12 contains a brief part description with index prefix = DESCRIBE; entering

(1) DESCRIBE EQ ANGLE

will result in retrieving all data records for angle discrete parts within the data base; or entering

(2) DESCRIBE EQ CONSTANT SECTION

will result in the retrieval of all constant section data records.

Figure 17 (Page 70) shows Demonstration MC/DG Index and Range Terms for retrieval usage.

3. SAMPLES OF USAGE--PHASE I

During the course of the MC/DG computerization concept validation study, several categories of samples of usage were developed. The purpose in the development of such samples of usage was threefold:

- Document several modes of usage of the computerized MC/DG in order to validate the success of our study
- Demonstrate the usage of macrolevel procedures (called PROFILES for the BASIS system) and their utility for novice users

- Demonstrate the utility of on-line training macro procedures as a part of the overall user training program.

a. BASIS Profiles

The samples of usage presented in this report demonstrate only a small fraction of the many ways that the user could utilize the computerized MC/DG. To save user time in entering repeated retrieval requests, a library of PROFILES was created. A sample of the current PROFILE library is illustrated in Figure 18 (Page 71).

A BASIS profile is a saved procedure consisting of a series of BASIS commands. The profiles can (and should) have descriptive text combined with the BASIS commands so that the user is notified of operations being performed or entries to be made in performance of these operations. Thus, PROFILES serve as a high level, English language programming mode. Trained users can create and save their own PROFILES. Novice users may simply utilize the existing PROFILE library. The contents of the PROFILE library may be displayed by entering the command, PROFILE SHOW (or, /$, as an abbreviation), see Figure 18 on Page 71.

b. Sample User Sessions

Three sample user sessions are described in this section. These sessions are merely illustrative of a large variety of user sessions which make use of the PROFILES. In particular, each of the three sample sessions illustrates:

- Data retrieval
- Tabulation of retrieved data
- Display of retrieved data.

The mode of accomplishing the basic functions of retrieval, tabulation, and display varies among the sample sessions. This variation is shown to illustrate the relative advantages of one functional mode over others.

c. Session 1 - Qualitative Cost Driver Effect Data

The first sample session, Figure 19 (Page 72), illustrates how the user can obtain qualitative cost driver effect (CDE) data via the following steps:

(1) Log in for the computer system

(2) Log on for the BASIS system; BASIS responds by listing the data base being used, date of last update, and the current number of data base records

(3) Select and execute a PROFILE named COMPARISON BAR CHARTS

(4) Observe the prompting messages issued by this PROFILE regarding the entries to be made

(5) Make entries to control the retrieval, tabulation, and display modes within the PROFILE; the sample session is for lineal aluminum part 2B, length = 24 inches, to compare the BPCOST and JOGGLES cost (cost of the base part plus joggles) as a function of the cost-driver manufacturing method

(6) Enter the BASIS LIST command to itemize the retrieval commands submitted by the PROFILE to the BASIS retrieval module; the result is that three records were retrieved by the five retrieval commands

(7) Examine the tabulation of the retrieved records; to achieve the cost comparison study for the brake and buffalo roll, brake and stretch, and rubber press manufacturing methods.

A bar chart of the man-hour costs for the base part, plus joggles for the three manufacturing methods, shows that the minimum cost (by the rubber press method) is about 0.3 man-hours per part (see Figure 20, Page 73). The entire user session was accomplished by making 10 user entries (system log in, BASIS log on, PROFILE selection, and the answer to seven questions). The session was completed in about 3 minutes time.

d. Session 2 - The Effect of Lot Size and Part Length on Part Cost

The second sample session illustrates a tabular and graphic display of the effect of lot size and part length on part cost. The following steps are illustrated in Figure 21 (Page 74):

(1) Log on for the BASIS system

(2) Select the PROFILE named SAMPLE USAGE which asks the user to select a material, part code, and manufacturing method

(3) Review the available choices of retrieval selection. For example, material choices are aluminum, steel, and titanium (other options on the list are redundant options such as aluminum, aluminum-2024, and 2024). By entering a simple letter choice from the list provided, the retrieval is accomplished. Advantages of this mode are that

- More than one choice from the list can be made (i.e., aluminum and steel could be selected)
- The available choices are explicitly shown.

(4) The main profile used in this session calls the BASIS SORT command to sort the retrieval records by part length.

(5) The main profile then calls "CALC LO, LI, L4, L5, L6, C1" to perform calculations and numeric data for tabulation.

(6) Finally, the PROFILE IDENTIFY was called to tabulate data elements from the retrieved records. This PROFILE issues commands to the BASIS report module named FORMAT. The FORMAT commands are suppressed by the OUTPUT=OFF specification on the PROFILE EXECUTE command.

A teletype plot of the man-hour cost data is made for the four lot sizes (5, 10, 25, and 50) as a function of part length (data for four part lengths--24, 48, 96, and 144 inches). The plot specifications are supplied by the main PROFILE. Despite the disadvantage of not being able to draw continuous lines, the teletype plot has the advantage that it can be done on any teletype-compatible terminal (CRT character mode and hard copy terminals) (see Figure 21, Page 74).

e. Session 3 - The Effect of Lot Size and Part Length

The third sample session shown in Figure 22 (Page 77) illustrates a tabular and graphic display of the effect of lot size and part length on part cost. The entries for PROFILE selection and retrieval selections are similar to those for Session 2:

(1) Tabulation of identifying data elements is achieved by the PROFILE named IDENTIFY, but in this session the FORMAT commands are listed.

(2) An X-Y plot of the retrieved data is performed on a graphic (storage tube) computer terminal. The commands for the PLOTXY (graphic terminal) module are similar to the commands for the LPLOT (teletype line plot) module.

4. ADVANCED COMPOSITES WITH POLYMER MATRICES--FACTORS TO BE CONSIDERED FOR INCLUSION IN MC/DG DATA BASE

The considerations for the advanced composites section of the MC/DG data base can be categorized into four sections:

- Material
- Shape
- Designer-Influenced Cost Elements (DICE)
- Manufacturing.

This report will present items to be considered utilizing the present advanced composites data and items to be considered in near-term follow-on MC/DG efforts. Allowances must be made in the design of the advanced composites data base for the expansion that will become necessary as new developments in the composites field are readied for industry utilization.

a. Material

The material considerations involve the fiber, matrix, and pre-combined fiber/matrix systems used to construct the composite parts. This area is one in which considerable expansion is envisioned, as new fibers, matrices, and available forms of these are developed and introduced into industry. One widely used variable is the fiber content of the composite system. Other considerations are as follows:

<u>For the Fibers</u>

- Type
  - Boron
  - Glass
  - Graphite
  - Kevlar
  - Possibly five others

- Form
  - Broadgoods (many possible weaves)
  - Chopped
  - Unidirectional

<u>For the Matrix</u>

- Type
  - Epoxy
  - Polyimide
  - Polysulfones
  - Possibly 10 others
- Form
  - Current production materials
  - Advanced, rapid-cure matrices

<u>For Fiber/Matrix Preforms</u>

- Bulk Molding Compounds (BMC)
- Preplied
- Sheet Molding Compounds (SMC)
- Tape (various widths)
- Combined SMC and Continuous Fiber Preforms.

Other items to consider are the ply thickness and the possibility of up to six different fibers being combined to form a hybrid composite.

b. Shape

The factors to be considered regarding shape include ply orientation, the number of plies used, and the number of basic shapes required to form a complex structural shape or assembly (e.g., two channels and a flat part to make an "I" section). Breaking this section into factors affecting lineal and sheet parts yields the following considerations:

<u>For Lineal Parts</u>

- Shape description
- Dimensions
- Bend radii
- Radius of curvature of the part
- Twist

For Sheet Parts

- Single curvature
- Compound curvature
- Dimensions
- Twist.

c. Designer-Influenced Cost Elements (DICE)

Once the designer has the basic shape of the part defined, he can then consider modifications to that shape which will better reflect the design requirements. Some of these modifications, referred to as Designer-Influenced Cost Elements (DICE) in the development of the MC/DG, are listed below:

- Joggles
- Edge doubler
- Stringer doubler
- Pad doubler
- Flanged lightening holes
- Cut-outs
- Integral shear clips
- Trim
- Inserts
  - Metal
  - Composite
- Tolerance
- Special surface protection
  - Lightning strike
  - Paint
  - Water proofing
  - Ultraviolet
- Machining
- Mechanical fasteners
  - Nut plates
  - Dzus fasteners
  - Toggles
  - Others.

d. Manufacturing

Manufacturing is another area in which great expansion is foreseen. With the cost of composite materials becoming lower, commercial, i.e., non-military, industries with high-quality products are beginning to consider composite materials for incorporation into their products. To effectively make use of composites in commercial products, manufacturing methods must be developed to facilitate the high production quantity. Work on this problem has just begun, and thus is not included explicitly in this report, other than to point out that room must be provided in the data base for expansion to include new methods developed by commercial industry. Considerations relative to manufacturing methods currently being used, and those likely to become feasible in the near future, are listed below:

- Hand layup versus automatic
- Manual versus automatic material cutting
- Cure
  - Cure stage (e.g., "B-stage" or fully cured)
  - Autoclave
  - Oven/vacuum bag
- Spray up
- Injection molding
- Matched die molding
- Filament winding
- Compression molding
- Integrally heated tools
- Thermoplastic composite forming
  - Continuous roll
  - Vacuum
  - Extrusion
  - Press molding
- Pultrusion.

The search for potential applications of composite materials has only recently started. As new uses for these materials are devised, new technology will be developed to incorporate composites into new products in a cost-competitive manner. For the MC/DG to be a tool that a designer

can effectively use, the data base must be designed so that it can be easily expanded to incorporate new technology as it develops.

5. MECHANICALLY-FASTENED ASSEMBLIES--FACTORS TO BE CONSIDERED FOR INCLUSION IN EXPANDED MC/DG DATA BASE

The candidate elements for the mechanically-fastened assemblies section of the MC/DG data base can be divided into four categories:

- Material
- Shape
- Fastener type
- Assembly method.

This report will present items to be considered using present mechanically-fastened assembly data as well as items to be considered in near-term follow-on MC/DG efforts.

a. Material

The material category includes the type of material used in the preassembled parts as well as the type of fastener material. Some considerations for this area include:

For the Preassembled Parts

- Aluminum 2024
- Titanium 6Al-4V
- Steel Ph15-7Mo

For the Fasteners

- Aluminum 2024
- Bimetallics.

b. Shape

The factors to be considered concerning assembly shape include overall assembly dimensions, skin panel thickness, type of panel stiffener (e.g., angle, channel, or z-shaped stiffeners), number of preassembled parts, and number of fasteners required to construct each assembly.

c. Fastener Type

The types of fasteners used to join the assembled parts can include:

- Countersunk rivets
- Universal head rivets
- Countersunk bolts
- Universal head bolts.

d. Assembly Method

The method of fastener installation falls into this category. In addition, since fasteners installed in fuel areas may require special treatment to prevent possible leaks, the considerations relative to assembly methods should reflect this requirement. The methods of fastener installation can include:

- Manual
- Automatic
- Combined automatic/manual.

The fastener sealant requirements can include:

- Fasteners installed dry
- Fasteners installed wet with a sealant or primer
- Fasteners installed wet with sealant or primer and faying surfaces sealed.

6. DESCRIPTION OF THE PHASE II DEMONSTRATION DATA BASE

a. Data Base Structure and Storage

The Phase II data base is structurally similar to the Phase I data base. Physically, it shares the same six data base files described in the preceeding section for the Phase I data base. The Source Data File was expanded to include data for mechanically-fastened assemblies and advanced composite fabrication. The Index and Range Files were enlarged to contain the added index terms and range terms. The Table File was reconstructed to accommodate the Phase II data.

Appendix A shows the new version of the DDL program for Phase II. Note the added data fields:

Field 23: NOFPARTS = number of parts

Field 24: NFASTENR = number of fasteners

Field 25: CASSCOST = complete assembly cost

Field 42: EDGEDBLR = edge doubler cost

Field 43: STGRDBLR = stringer doubler cost

Field 44: PADDBLR = pad doubler cost

Field 45: INSHCLIP = integral shear clips cost

Field 46: FASTYPE 1 = countersunk rivet cost

Field 47: FASTYPE 2 = universal head rivet cost

Field 48: FASTYPE 3 = countersunk Hi-Lok bolt cost

Field 49: FASTYPE 4 = universal Hi-Lok bolt cost

Field 50: FASTYPE 5 =

Field 130: PLYO = 0° ply count

Field 131: PLY45 = 45° ply count

Field 132: PLY 90 = 90° ply count

Field 133: STRIPO = 0° strip plies

Field 134: STRIP45 = 45° strip plies

Field 135: STRIP90 = 90° strip plies

Field 136: STRINGER = hat stringers cost

Field 137: FRAMES = frames cost.

The mode of Phase II data storage is the same as for Phase I.

b. Data Base Construction

The following subtasks were performed to construct the Phase II data base:

- Review of the Phase I data base structure
- Analysis of the Phase II data collection summary forms
- Review of data display formats required by Phase II
- Construction of the data base files
- Testing of data display formats.

(1) Review of the Phase I Data Base

Review of the Phase I data base was necessary to determine the design changes needed for processing the Phase II(a) and II(b) data. The DDL program for Phase I data base was expanded to include the definition

of the new data fields, their respective indexing prefixes, display formats, labels, and mapping functions. New formats of query for textual data index and numeric data range search were implemented in the Indexing Definition paragraph.

(2) Analysis of the Data Collection Summary Forms

Analysis of the data collection summary forms was needed for the design of coding and keying data in preparation of its presentation to the data base. A new preprocessing program was coded to decode the data items from punched cards. Figure 23 (Page 80) shows the printed output of this program for the data fields in mechanically-fastened assembly data record while Figure 24 (Page 82) shows those for a composite part data record.

(3) Review of Data Display Formats

Review of data display formats was done to determine the conceptual design of REPORT and PLOT module program changes.

(4) Construction of the Data Base Files

Data in the printed output from the preprocessing program was randomly selected for checking against those in the data collection summary forms. Errors were corrected and re-preprocessed. All acceptable source data were input to an Input Processor program to be transformed into "transactions" for input to the Source Data File. The Index File Manager and Range File Manager in the BASIS file maintenance module completed the insertion of the new index and range terms into the Index and Range Files.

The new version of the Table File was built when the revised DDL program was recompiled.

(5) Testing of Data Display Formats

Testing of data display formats was done on various sets of Phase II data. As was reported in MM Status Report No. 12, it was determined that several of the display formats, such as the nomograms, could not be computerized due to present graphics software package limitations. Also, several sample formats could not be computerized for lack of data, i.e., comparison of cost of wet and dry fastener installation

are not included in the Phase II(a) data base. It was determined that Phase II display format samples of usage would be limited to tables, bar graphs, and x-y plots. As a substitute for the nomographs (which are somewhat complex), it was decided by the computerization staff that a computerized question-and-answer session could reduce the complexity of the display format and result in a graph that could be displayed using the available graphics software.

The nomograph technique is often used for representing data which are dependent upon several independent variables. In the case of the lineal part advanced composite fabrication man-hour cost data, costs are dependent upon part length, total number of plies, and developed width (flat pattern) of the part. For reference, the nomograph for the "I" section lineal composite part recurring cost is shown in Figure 25 (Page 85). The technique used for the computerized Phase II composite part data is to ask the user to specify a choice of length, ply count, or width as the primary cost variable of interest; the user is then asked to pick fixed values for the other two variables. A plot of base part recurring or non-recurring tooling cost versus the chosen cost variable is automatically graphed for the user.

7. SAMPLES OF USAGE

Concurrent with the task on data display format testing, several samples of Phase II usage have been prepared. An example of one such sample of usage for the "I" section lineal composite part is illustrated in Figure 26 (Page 86). In this sample, the user has made the following entries:

(1) /LOGON, BASIS, PHASE II--This entry initiates usage of the computerized data base.

(2) PROFILE EXECUTE START COMPOSITE--This entry initiates a sequence of stored procedures (PROFILES) which request user choices of options (discussed below).

(3) PART SHAPE?I--In response to the question of what part shape is desired, the user has entered I (for I section lineal); valid responses are: I, J. HAT, and SKIN.

(4) DRIVER?LENGTH--In response to the question about driver (primary cost variable of interest), the user has selected part length; valid responses are: LENGTH, DEV WIDTH (developed width), TOTAL PLIES, and AREA.

(5) PLY COUNT:ALL--In response to the question about ply count, the user has selected ALL ply; valid responses are: 10, 20, 32, and ALL.

(6) PART WIDTH?3--In response to the question on part width, the user has selected 3 inches (parts 3.00 to 3.99 inches wide); valid responses are: 3, 4, 5, 6, 9, and ALL.

(7) PART SHAPE?I--See earlier remarks.

(8) COST ELEM?BPCOST--The cost element selected is a user choice of BPCOST (base part cost) or NRTCOST (non-recurring tooling cost).

Another sample of usage for mechanically-fastened assemblies is illustrated in Figure 27 (Page 89). In this sample, the user had made the following request entries:

(1) BASIS, DEMO, MCDG--This entry calls up the BASIS software system and the MC/DG data base.

(2) PROFILE EXECUTE START ASSEMBLY--This entry starts the execution of a saved procedure (PROFILE) named START ASSEMBLY.

(3) TITANIUM--The user is asked to supply a parameter to select the material type used for the assembly (the data base contains aluminum and titanium assemblies).

(4) The user is then requested to select from two lists of options. DESCRIBE lists three choices for the type of assembly--AVIONICS, BAY, and DOOR; Option C, DOOR, was selected by the user. The second option list is for INSTMETH (installation method); here the choices AUTOMATIC and MANUAL were made by the entry of B,D.

The remaining sample of usage was generated by the same procedure. A table lists assembly size, number of parts, fastener count for types 1-4, total fastener count, base assembly man-hours, total assembly man-hours, and non-recurring tooling man-hours. A graph of cost (man-hours) per fastener versus the number of fasteners per assembly is plotted as a teletype scatter diagram.

These samples of usage utilize the previously described question-and-answer technique designed as an interim step for displaying nomograph information. Further development of format data display software will be necessary to allow the designer more flexibility.

8. DEVELOPMENT OF A USERS GUIDE

A manual describing the use of the computerized MC/DG is required, to provide more detailed operating instructions and module descriptions than is possible in the on-line tutorials. This users guide should be simple to use and yet comprehensive to the novice MC/DG user, but also should provide sufficient information for a more experienced user to be able to create his own modules and PROFILES. An example of a users guide for the demonstration MC/DG system is included as Appendix D.

FIGURE 14. EXAMPLE OF DATA SUMMARY SHEET

Part No.:
Material Type and Specification:
Material Final Condition:
Manufacturing Method:
Airframe Company Source:
Date Data Calculated:
Brief Part Description:
Part Geometry: Straight ____________ or Curved ____________ Radius ____________ inches

	Dimensions, inches			Base Part Cost (M-H)	Non-Recurring Tooling Cost (M-H)	Designer-Influenced Cost Elements (DICE)									Discrete Part Cost (M-H)
Lot Size	Length					A	B	C	D	E	F	G	H	I	
5															
10															
25															
50															

D A T A   S U M M A R Y   S H E E T

06/30/7

PART NO: MC/DG - A - 2P
MATERIAL TYPE AND SPECIFICATION: ALUMINUM-2024
MATERIAL FINAL CONDITION: T62
MANUFACTURING METHOD: BRAKE AND STRETCH
BRIEF PART DESCRIPTION: CURVED CHANNEL CONSTANT SECTION
PART GEOMETRY: STRAIGHT: CURVED:

** A V E R A G E S **

LOTSIZE	LENGTH	JOGGLES	L.HOLES	BASE PART COST	TOTAL	N.R.T. COSTS			COMPLEXITIES COSTS A	B	D	G	DIS. PART COST
5	24.0	2.0	2.0	.796	.453	.349	.054	0.000	.082	.104	.178	.107	1.720
5	48.0	2.0	3.0	.920	.517	.405	.067	0.000	.086	.117	.251	.146	2.037
5	96.0	2.0	5.0	1.348	.693	.554	.112	0.000	.110	.159	.420	.231	2.961
5	144.0	2.0	7.0	1.534	.867	.703	.153	0.000	.114	.188	.561	.319	3.583
10	24.0	2.0	2.0	.541	.453	.349	.054	0.000	.056	.062	.145	.086	1.343
10	48.0	2.0	3.0	.650	.517	.405	.067	0.000	.059	.075	.217	.124	1.643
10	96.0	2.0	5.0	.997	.693	.554	.112	0.000	.084	.122	.383	.209	2.488
10	144.0	2.0	7.0	1.196	.867	.703	.153	0.000	.089	.146	.525	.293	3.117
25	24.0	2.0	2.0	.389	.453	.349	.054	0.000	.040	.039	.125	.073	1.119
25	48.0	2.0	3.0	.438	.517	.405	.067	0.000	.044	.051	.197	.110	1.407
25	96.0	2.0	5.0	.799	.693	.554	.112	0.000	.068	.093	.362	.195	2.210
25	144.0	2.0	7.0	.996	.867	.703	.153	0.000	.072	.122	.503	.277	2.837
50	24.0	2.0	2.0	.337	.453	.349	.054	0.000	.035	.030	.118	.069	1.043
50	48.0	2.0	3.0	.433	.517	.405	.067	0.000	.039	.042	.191	.106	1.328
50	96.0	2.0	5.0	.738	.693	.554	.112	0.000	.063	.085	.354	.191	2.123
50	144.0	2.0	7.0	.928	.867	.703	.153	0.000	.067	.113	.496	.272	2.744

FIGURE 15. DEMONSTRATION MC/DG PHASE I DATA SUMMARY REPORT

DATA SUMMARY SHEET

PART NO: MC/DG - A - 25 06/30/7
MATERIAL TYPE AND SPECIFICATION: ALUMINUM-2024
MATERIAL FINAL CONDITION: T62
MANUFACTURING METHOD: BRAKE + BUFFALO ROLL
BRIEF PART DESCRIPTION: CURVED CHANNEL CONSTANT SECTION
PART GEOMETRY: STRAIGHT: CURVED:

** A V E R A G E S **

				BASE PART		N.R.T.	COSTS			COMPLEXITIES COSTS			DIS. PART
LOTSIZE	LENGTH	JOGGLES	L.HOLES	COST	TOTAL				A	B	D	G	COST
5	24.0	2.0	2.0	.631	.330	.223	.078	0.000	.082	.111	.226	.123	1.502
5	48.0	2.0	3.0	.746	.339	.226	.085	0.000	.065	.119	.338	.154	1.782
5	96.0	2.0	5.0	1.169	.367	.238	.107	0.000	.092	.135	.555	.219	2.547
5	144.0	2.0	7.0	1.457	.394	.249	.128	0.000	.098	.151	.800	.292	3.191
10	24.0	2.0	2.0	.423	.330	.223	.078	0.000	.048	.067	.157	.097	1.152
10	48.0	2.0	3.0	.498	.339	.226	.085	0.000	.053	.075	.299	.124	1.384
10	96.0	2.0	5.0	.874	.367	.238	.145	0.000	.058	.090	.526	.193	2.108
10	144.0	2.0	7.0	1.154	.394	.249	.128	0.000	.064	.102	.762	.251	2.726
25	24.0	2.0	2.0	.298	.330	.223	.078	0.000	.025	.040	.164	.081	.938
25	48.0	2.0	3.0	.380	.339	.226	.085	0.000	.027	.048	.276	.109	1.178
25	96.0	2.0	5.0	.698	.367	.238	.107	0.000	.034	.064	.513	.174	1.541
25	144.0	2.0	7.0	.974	.394	.249	.128	0.000	.040	.080	.738	.240	2.465
50	24.0	2.0	2.0	.261	.330	.223	.078	0.000	.019	.033	.156	.076	.875
50	48.0	2.0	3.0	.341	.339	.226	.085	0.000	.021	.041	.268	.101	1.111
50	96.0	2.0	5.0	.643	.367	.238	.107	0.000	.029	.057	.496	.169	1.760
50	144.0	2.0	7.0	.916	.394	.249	.128	0.000	.033	.072	.771	.235	2.381

FIGURE 15. (Continued)

DATA SUMMARY SHEET

06/30/7

PART NO: MC/DG - S - 28
MATERIAL TYPE AND SPECIFICATION: STEEL - PH15 - 7MO
MATERIAL FINAL CONDITION: TH1050
MANUFACTURING METHOD: COLD BRAKECOLD STFT.
BRIEF PART DESCRIPTION: CURVED CHANNEL CONSTANT SECTION
PART GEOMETRY: STRAIGHT: CURVED:

** A V E R A G E S **

LOTSIZE	LENGTH	JOGGLES	L.HOLES	BASE PART COST	TOTAL	N.R.T.	COSTS		A	B	G	DIS. PART COST
									COMPLEXITIES COSTS			
5	24.0	2.0	2.0	.944	.563	.481	.138	0.000	.126	.127	.329	2.088
5	48.0	2.0	3.0	1.108	.664	.565	.165	0.000	.129	.143	.423	2.466
5	96.0	2.0	5.0	1.819	.874	.747	.213	0.000	.131	.176	.633	3.633
5	144.0	2.0	7.0	2.282	1.064	.893	.287	0.000	.177	.254	.847	4.624
10	24.0	2.0	2.0	.750	.563	.481	.138	0.000	.085	.079	.253	1.740
10	48.0	2.0	3.0	.948	.664	.565	.165	0.000	.088	.094	.358	2.151
10	96.0	2.0	5.0	1.586	.874	.747	.213	0.000	.090	.125	.568	3.243
10	144.0	2.0	7.0	2.035	1.064	.893	.287	0.000	.135	.201	.782	4.217
25	24.0	2.0	2.0	.632	.563	.481	.138	0.000	.060	.050	.220	1.526
25	48.0	2.0	3.0	.825	.664	.566	.165	0.000	.063	.054	.319	1.936
25	96.0	2.0	5.0	1.430	.874	.747	.213	0.000	.065	.094	.529	3.000
25	144.0	2.0	7.0	1.886	1.064	.893	.287	0.000	.112	.168	.742	3.972
50	24.0	2.0	2.0	.598	.563	.481	.138	0.000	.052	.041	.202	1.456
50	48.0	2.0	3.0	.784	.664	.565	.165	0.000	.055	.055	.306	1.864
50	96.0	2.0	5.0	1.390	.874	.747	.213	0.000	.057	.084	.515	2.921
50	144.0	2.0	7.0	1.836	1.064	.893	.287	0.000	.071	.157	.729	3.857

FIGURE 15. (Continued)

DATA SUMMARY SHEET

PART NO: MC/DG - A - 3B 06/30/7
MATERIAL TYPE AND SPECIFICATION: ALUMINUM-2024
MATERIAL FINAL CONDITION: T62
MANUFACTURING METHOD: BRAKE + BUFFALO ROLL
BRIEF PART DESCRIPTION: CURVED ZEE CONSTANT SECTION
PART GEOMETRY: STRAIGHT: CURVED:

** A V E R A G E S **

LOTSIZE	LENGTH	JOGGLES	L.HOLES	BASE PART COST	TOTAL	N.R.T.	COSTS		COMPLEXITIES COSTS A	B	D	G	DIS. PART COST
5	24.0	3.0	2.0	.572	.321	.267	.067	0.000	.122	.104	.215	.105	1.439
5	48.0	4.0	3.0	.695	.336	.272	.080	0.000	.135	.118	.317	.148	1.748
5	96.0	6.0	5.0	1.131	.379	.287	.115	0.000	.223	.160	.552	.226	2.673
5	144.0	8.0	7.0	1.326	.425	.306	.149	0.000	.271	.188	.754	.312	3.276
10	24.0	3.0	2.0	.377	.321	.267	.067	0.000	.082	.063	.183	.080	1.105
10	48.0	4.0	3.0	.498	.336	.272	.080	0.000	.095	.075	.284	.125	1.413
10	96.0	6.0	5.0	.866	.379	.287	.115	0.000	.183	.117	.516	.205	2.270
10	144.0	8.0	7.0	1.056	.425	.306	.149	0.000	.231	.145	.718	.290	2.866
25	24.0	3.0	2.0	.283	.321	.267	.067	0.000	.056	.039	.163	.064	.926
25	48.0	4.0	3.0	.378	.336	.272	.080	0.000	.069	.050	.265	.114	1.212
25	96.0	6.0	5.0	.712	.379	.287	.115	0.000	.157	.092	.495	.194	2.029
25	144.0	8.0	7.0	.896	.425	.306	.149	0.000	.205	.120	.696	.278	2.622
50	24.0	3.0	2.0	.251	.321	.267	.067	0.000	.049	.031	.156	.060	.868
50	48.0	4.0	3.0	.343	.336	.272	.080	0.000	.062	.043	.258	.110	1.152
50	96.0	6.0	5.0	.661	.379	.287	.115	0.000	.150	.085	.487	.190	1.953
50	144.0	8.0	7.0	.827	.425	.306	.149	0.000	.199	.113	.689	.275	2.527

FIGURE 15. (Continued)

D A T A   S U M M A R Y   S H E E T

PART NO: MC/DG - A - 3B 06/30/7
MATERIAL TYPE AND SPECIFICATION: ALUMINUM-2024
MATERIAL FINAL CONDITION: T62
MANUFACTURING METHOD: BRAKE AND STRETCH
BRIEF PART DESCRIPTION: CURVED ZEE CONSTANT SECTION
PART GEOMETRY: STRAIGHT: CURVED:

** A V E R A G E S **

				BASE PART		N.R.T. COSTS			COMPLEXITIES COSTS				DIS. PART
LOTSIZE	LENGTH	JOGGLES	L.HOLES	COST	TOTAL				A	B	D	G	COST
5	24.0	3.0	2.0	.800	.526	.485	.069	0.000	.100	.106	.182	.112	1.827
5	48.0	4.0	3.0	.964	.599	.548	.086	0.000	.114	.129	.257	.148	2.201
5	96.0	6.0	5.0	1.404	.794	.714	.133	0.000	.204	.162	.438	.228	3.230
5	144.0	8.0	7.0	1.633	.983	.876	.179	0.000	.252	.191	.594	.312	3.964
10	24.0	3.0	2.0	.550	.526	.485	.069	0.000	.072	.065	.153	.093	1.459
10	48.0	4.0	3.0	.684	.599	.548	.086	0.000	.086	.077	.228	.125	1.799
10	96.0	6.0	5.0	1.048	.794	.714	.133	0.000	.176	.120	.405	.205	2.749
10	144.0	8.0	7.0	1.271	.983	.876	.179	0.000	.224	.152	.560	.290	3.481
25	24.0	3.0	2.0	.403	.526	.485	.069	0.000	.053	.041	.133	.081	1.238
25	48.0	4.0	3.0	.521	.599	.548	.086	0.000	.067	.052	.209	.114	1.562
25	96.0	6.0	5.0	.898	.794	.714	.133	0.000	.157	.095	.394	.194	2.522
25	144.0	8.0	7.0	1.054	.983	.876	.179	0.000	.205	.124	.539	.278	3.183
50	24.0	3.0	2.0	.351	.526	.485	.069	0.000	.049	.033	.127	.077	1.164
50	48.0	4.0	3.0	.464	.599	.548	.086	0.000	.063	.044	.202	.110	1.482
50	96.0	6.0	5.0	.762	.794	.714	.133	0.000	.153	.087	.376	.190	2.363
50	144.0	8.0	7.0	.981	.983	.876	.179	0.000	.200	.115	.532	.275	3.087

FIGURE 15. (Continued)

RECORD 81357

```
81357 190081357
81357 2900GROUP TECH
81357 3900MC-DS DATA
81357 4900AVGERAGES
81357 5900    3A
81357 6900ALUMINUM-2024
81357 7900T62
81357 8900BRAKE FORM
81357 990004/11/78
8135710900THIS IS THE STATISTICS RECORD
8135711900010
8135712900STRAIGHT ZEE CONSTANT SECTION; STIFFENER,STRINGER
8135713900
8135714900
8135715900
8135716900    144.000
8135717900      0.000
8135718900      0.000
8135719900      0.000
8135720900       .063
8135721900      8.000
8135722900      7.000
8135723900      0.000
8135724900      0.000
8135725900      0.000
8135726900       .559
8135727900      0.000
8135728900      0.000
8135729900      0.000
8135730900      1.027
8135731900       .207
8135732900       .267
8135733900       .170
8135734900      0.000
8135735900      1.835
8135736900      0.000
8135737900      0.000
8135738900      0.000
8135739900      0.000
8135740900      0.000
8135741900      3.506
```

FIGURE 16. PHASE I INPUT PREPROCESSING PROGRAM PRINTOUT

```
RECORD  81358

81358 190081358
81358 2900GROUP TECH
81358 3900MC-DG DATA
81358 4900AVGERAGES
81358 5900    3A
81358 6900ALUMINUM-2024
81358 7900T62
81358 8900BRAKE FORM
81358 990004/11/78
8135810900THIS IS THE STATISTICS RECORD
813581190025
8135812900STRAIGHT ZEE CONSTANT SECTION; STIFFENER,STRINGER
8135813900
8135814900
8135815900
8135816900     24.000
8135817900      0.000
8135818900      0.000
8135819900      0.000
8135820900       .063
8135821900      3.000
8135822900      2.000
8135823900      0.000
8135824900      0.000
8135825900      0.000
8135826900       .159
8135827900      0.000
8135828900      0.000
8135829900      0.000
8135830900       .268
8135831900       .171
8135832900       .063
8135833900       .048
8135834900      0.000
8135835900       .288
8135836900      0.000
8135837900      0.000
8135838900      0.000
8135839900      0.000
8135840900      0.000
8135841900       .838
```

FIGURE 16. (Continued)

TABLE 1. MC/DG DEMONSTRATION DATA BASE DATA ELEMENTS

Data Element Number	Data Element Name	Data Element Description
001	ACC	ACCESSION NUMBER
002	GROUPTEC	GROUP TECHNOLOGY
003	FILE	FILE NAME
004	SUBFILE	SUBFILE NAME
005	PARTCODE	DISCRETE PART CODE
006	MATERIAL	MATERIAL USED
007	MATFINAL	MATERIAL FINAL CONDITION
008	MFGMETH	MANUFACTURING METHOD
009	DATADATE	DATE DATA GENERATED
010	DATATYPE	TYPE OF DATA SUBFILE
011	LOTSIZE	MANUFACTURING LOT SIZE
012	DESCNUSE	BRIEF PART DESCRIPTION
016	LENGTH	PART MEASUREMENT 1, LENGTH
017	WIDTH	PART MEASUREMENT 2, WIDTH
018	RADIUS	PART MEASUREMENT 3, RADIUS
019	HEIGHT	PART MEASUREMENT 4, HEIGHT
020	THICK	PART MEASUREMENT 5, THICKNESS
021	NJOGGLES	NUMBER OF JOGGLES
022	NFLHOLES	NUMBER OF FLANGED HOLES
026	BPCOST	BASE PART COST
027	BPNRCOST	BASE PART NON-RECUR TOOL COST
028	LTNRCOST	LIN. TRIM NON-RECUR TOOL COST
029	ETNRCOST	END TRIM NON-RECUR TOOL COST
031	NRTCOST	TOTAL NON-RECURRING TOOL COST
032	JOGGLES	JOGGLES COST
033	FLHOLES	FLANGED HOLES COST
034	BEADS	BEADS COST
035	HTREAT	HEAT TREATMENT COST
036	SURFIN	SURFACE FINISH COST
037	TOLERAN	TOLERANCE COST
038	LINTRIM	LINEAL TRIM COST
039	ENDTRIM	END TRIM COST
040	UNFLHOLE	CUT-OUTS WITHOUT FLANGES COST
041	DPCOST	DISCRETE PART COST
050	BPCOSTC0	BASE PART COST REG. COEF. C0
051	BPCOSTC1	BASE PART COST REG. COEF. C1
052	BPCOSTC2	BASE PART COST REG. COEF. C2
053	BPCOSTC3	BASE PART COST REG. COEF. C3
054	BPCOSTC4	BASE PART COST REG. COEF. C4
055	BPCOSTC5	BASE PART COST REG. COEF. C5
056	BPCOSTC6	BASE PART COST REG. COEF. C6

TABLE 1. (Continued)

Data Element Number	Data Element Name	Data Element Description
057	BPCOSTC7	BASE PART COST REG. COEF. C7
060	NRTC0	NON-REC TOOL COST REG COEF C0
061	NRTC1	NON-REC TOOL COST REG COEF C1
062	NRTC2	NON-REC TOOL COST REG COEF C2
063	NRTC3	NON-REC TOOL COST REG COEF C3
064	NRTC4	NON-REC TOOL COST REG COEF C4
065	NRTC5	NON-REC TOOL COST REG COEF C5
066	NRTC6	NON-REC TOOL COST REG COEF C6
067	NRTC7	NON-REC TOOL COST REG COEF C7
070	JOGGLEC0	JOGGLES COST REG COEF C0
071	JOGGLEC1	JOGGLES COST REG COEF C1
072	JOGGLEC2	JOGGLES COST REG COEF C2
073	JOGGLEC3	JOGGLES COST REG COEF C3
074	JOGGLEC4	JOGGLES COST REG COEF C4
075	JOGGLEC5	JOGGLES COST REG COEF C5
076	JOGGLEC6	JOGGLES COST REG COEF C6
077	JOGGLEC7	JOGGLES COST REG COEF C7
080	FLHOLEC0	FLANGED HOLE COST REG COEF C0
081	FLHOLEC1	FLANGED HOLE COST REG COEF C1
082	FLHOLEC2	FLANGED HOLE COST REG COEF C2
083	FLHOLEC3	FLANGED HOLE COST REG COEF C3
084	FLHOLEC4	FLANGED HOLE COST REG COEF C4
085	FLHOLEC5	FLANGED HOLE COST REG COEF C5
086	FLHOLEC6	FLANGED HOLE COST REG COEF C6
087	FLHOLEC7	FLANGED HOLE COST REG COEF C7
090	HTREATC0	HEAT TREAT COST REG COEF C0
091	HTREATC1	HEAT TREAT COST REG COEF C1
092	HTREATC2	HEAT TREAT COST REG COEF C2
093	HTREATC3	HEAT TREAT COST REG COEF C3
094	HTREATC4	HEAT TREAT COST REG COEF C4
095	HTREATC5	HEAT TREAT COST REG COEF C5
096	HTREATC6	HEAT TREAT COST REG COEF C6
097	HTREATC7	HEAT TREAT COST REG COEF C7
100	LTRIMC0	LINEAL TRIM COST REG COEF C0
101	LTRIMC1	LINEAL TRIM COST REG COEF C1
102	LTRIMC2	LINEAL TRIM COST REG COEF C2
103	LTRIMC3	LINEAL TRIM COST REG COEF C3
104	LTRIMC4	LINEAL TRIM COST REG COEF C4
105	LTRIMC5	LINEAL TRIM COST REG COEF C5
106	LTRIMC6	LINEAL TRIM COST REG COEF C6
107	LTRIMC7	LINEAL TRIM COST REG COEF C7

TABLE 1. (Continued)

Data Element Number	Data Element Name	Data Element Description
110	ETRIMC0	END TRIM COST REG COEF C0
111	ETRIMC1	END TRIM COST REG COEF C1
112	ETRIMC2	END TRIM COST REG COEF C2
113	ETRIMC3	END TRIM COST REG COEF C3
114	ETRIMC4	END TRIM COST REG COEF C4
115	ETRIMC5	END TRIM COST REG COEF C5
116	ETRIMC6	END TRIM COST REG COEF C6
117	ETRIMC7	END TRIM COST REG COEF C7
120	UFHOLEC0	UNFLANGED HOLES COST REG C0
121	UFHOLEC1	UNFLANGED HOLES COST REG C1
122	UFHOLEC2	UNFLANGED HOLES COST REG C2
123	UFHOLEC3	UNFLANGED HOLES COST REG C3
124	UFHOLEC4	UNFLANGED HOLES COST REG C4
125	UFHOLEC5	UNFLANGED HOLES COST REG C5
126	UFHOLEC6	UNFLANGED HOLES COST REG C6
127	UFHOLEC7	UNFLANGED HOLES COST REG C7

```
 99  ACC:100000000/199999999
 96  ACC:300000000/399999999
 38  ACC:400000000/499999999
 36  ACC:600000000/699999999
292  ACC:700000000/799999999
 96  ACC:800000000/899999999
164  AREA:1000.001/3000.000
 79  AREA:144.001/240.000
125  AREA:240.001/480.000
 78  AREA:3000.001/5000.000
 36  AREA:480.001/524.000
 12  AREA:5000.001/99999.000
198  AREA:524.001/1000.000
 11  BPCOST:1.001/2.000
138  BPCOST:10.001/15.000
 82  BPCOST:15.001/20.000
 28  BPCOST:2.001/3.000
 46  BPCOST:20.001/40.000
 36  BPCOST:3.001/4.000
 76  BPCOST:4.001/5.000
  1  BPCOST:40.001/60.000
274  BPCOST:5.001/10.000
  4  CASCOST:1.001/2.000
 47  CASCOST:10.001/15.000
 23  CASCOST:15.001/20.000
 13  CASCOST:2.001/3.000
 27  CASCOST:20.001/40.000
 21  CASCOST:3.001/4.000
 29  CASCOST:4.001/5.000
  5  CASCOST:40.001/60.000
102  CASCOST:5.001/10.000
  1  CASCOST:60.001/80.000
184  DATADATE:781000/781099
508  DATADATE:781100/781199
272  DESCRIBE:ASSEMBLY
 88  DESCRIBE:AVIONICS
 88  DESCRIBE:BAY
420  DESCRIBE:COMPOSITE
384  DESCRIBE:CONSTANT
 36  DESCRIBE:CURVATURE
  4  DESCRIBE:CURVED
184  DESCRIBE:DOOR
272  DESCRIBE:FASTENED
152  DESCRIBE:HAT
116  DESCRIBE:I
116  DESCRIBE:J
272  DESCRIBE:MECHANICALLY
308  DESCRIBE:PANEL
420  DESCRIBE:PART
384  DESCRIBE:SECTION
 36  DESCRIBE:SINGLE
 36  DESCRIBE:SKIN
380  DESCRIBE:STRAIGHT
181  FASTYPE1:50/500
 84  FASTYPE1:501/999999
 16  FASTYPE2:15/19
 83  FASTYPE2:5/9
112  FASTYPE2:50/500
 45  FASTYPE3:10/14
 16  FASTYPE3:15/19
 16  FASTYPE3:5/9
 70  FASTYPE3:50/500
 32  FASTYPE4:15/19
 16  FASTYPE4:50/500
542  FILE:MC-DG DATA
100  INSTMETH:AUTO
 72  INSTMETH:AUTOMATIC
100  INSTMETH:MAN
100  INSTMETH:MANUAL

140  LENGTH:12.001/24.000
132  LENGTH:144.001/240.000
300  LENGTH:36.001/48.000
120  LENGTH:72.001/96.000
140  MATERIAL:ALUMINUM
140  MATERIAL:ALUMINUM-2024
420  MATERIAL:AS
420  MATERIAL:AS/3501-6 RESIN CONTENT 45+-3%
420  MATERIAL:CONTENT
420  MATERIAL:RESIN
132  MATERIAL:TITANIUM
132  MATERIAL:TITANIUM - 6AL - 4V
140  MATERIAL:2024
420  MATERIAL:3%
420  MATERIAL:3501
132  MATERIAL:4V
420  MATERIAL:45+
420  MATERIAL:6
132  MATERIAL:6AL
272  MFGMETH:=
 36  MFGMETH:ADHESIVE
308  MFGMETH:ASSEMBLY
 72  MFGMETH:ASSEMBLY METHOD = AUTOMATIC
100  MFGMETH:ASSEMBLY METHOD = COMBINED
100  MFGMETH:ASSEMBLY METHOD = MANUAL
420  MFGMETH:AUTOCLAVE
 72  MFGMETH:AUTOMATIC
 36  MFGMETH:BONDED
100  MFGMETH:COMBINED
384  MFGMETH:COMPONENTS
384  MFGMETH:CURING
420  MFGMETH:HAND
 36  MFGMETH:HAND LAYUP/AUTOCLAVE ADHESIVE-BONDED ASSEMBLY
384  MFGMETH:HAND LAYUP/AUTOCLAVE CURING COMPONENTS
420  MFGMETH:LAYUP
100  MFGMETH:MANUAL
272  MFGMETH:METHOD
167  NFASTENR:50/500
 60  NFASTENR:501/999999
 86  NOFPARTS:10/14
 50  NOFPARTS:15/19
 32  NOFPARTS:25/49
 40  NOFPARTS:5/9
 64  NOFPARTS:50/500
103  NRTCOST:100.001/200.000
  5  NRTCOST:1000.001/1100.000
  3  NRTCOST:1100.001/1200.000
  4  NRTCOST:1200.001/1300.000
  2  NRTCOST:1600.001/1700.000
  1  NRTCOST:1800.001/1900.000
  4  NRTCOST:1900.001/2000.000
125  NRTCOST:200.001/300.000
  2  NRTCOST:2000.001/99999.990
163  NRTCOST:300.001/400.000
 88  NRTCOST:400.001/500.000
 79  NRTCOST:500.001/600.000
 13  NRTCOST:600.001/700.000
  6  NRTCOST:700.001/800.000
  6  NRTCOST:800.001/900.000
 11  NRTCOST:900.001/1000.000
 44  PARTCODE:IIA- AL - 1
 52  PARTCODE:IIA- AL - 2
 44  PARTCODE:IIA- AL - 3
 44  PARTCODE:IIA- TI - 1
 44  PARTCODE:IIA- TI - 2
 44  PARTCODE:IIA- TI - 3
152  PARTCODE:IIB - C - 1
116  PARTCODE:IIB - C - 2

116  PARTCODE:IIB - C - 3
 36  PARTCODE:IIB - C - 4
 88  PARTFORM:AVIONICS BAY PANEL
420  PARTFORM:COMPOSITE PART
184  PARTFORM:DOOR PANEL
272  PARTFORM:MECHANICALLY FASTENED ASSEMBLY
 12  PLY0:1/2
  1  PLY0:10/14
235  PLY0:15/19
 40  PLY0:3/4
132  PLY0:5/9
 12  PLY45:1/2
116  PLY45:15/19
 40  PLY45:3/4
252  PLY45:5/9
116  PLY90:-500/0
 40  PLY90:1/2
132  PLY90:3/4
132  PLY90:5/9
 36  STRIP0:-500/0
 28  STRIP0:1/2
240  STRIP0:3/4
116  STRIP0:5/9
396  STRIP45:-500/0
 12  STRIP45:15/19
 12  STRIP45:5/9
420  STRIP90:-500/0
177  SUBFILE:AVGERAGES
515  SUBFILE:RAW DATA
  4  USAGE:CURVED J CONSTANT SECTION
 36  USAGE:SKIN PANEL SINGLE CURVATURE
152  USAGE:STRAIGHT HAT CONSTANT SECTION
116  USAGE:STRAIGHT I CONSTANT SECTION
112  USAGE:STRAIGHT J CONSTANT SECTION
132  WIDTH:24.101/36.100
338  WIDTH:3.101/6.100
 36  WIDTH:36.101/48.100
 46  WIDTH:6.101/9.100
 74  WIDTH:60.101/72.100
 66  WIDTH:84.101/96.100
```

FIGURE 17. DEMONSTRATION MC/DG INDEX AND RANGE TERMS FOR RETRIEVAL USAGE

```
PROFILE SHOW

  **** AVAILABLE PROFILES ****

  ****** SAVE FILE ******
ID        .NAME.
A1    SELECT
A2    START
A3    ANALYSIS
A4    POWER
A5    ASYMPTOTIC
A6    ANALYSIS PANEL
A7    POWER PANEL
A8    ASYMPTOTIC PANEL
A9    COSTS TRANSFORM
A10   RESIDUALS
A11   CALC COSTS VS LOTSIZE
A12   NEW LOTSIZE
A13   ANALYSIS LOTSIZE
A14   SELECT VS LOTSIZE
A15   PLOT VS LOTSIZE
A16   SELECT VS LENGTH, LOTSIZE
A17   CALC COSTS VS LENGTH, LOTSIZE
A18   ANALYSIS LENGTH, LOTSIZE
A19   SELECT COMPANY
A20   TAB CALC COSTS VS LENGTH, LOTS
A21   RESIUDALS LENGTH, LOTSIZE
A22   POWER VS LENGTH, LOTSIZE
A23   NEW C3
A24   LINEAR VS LENGTH, LOTSIZE
A25   TAB STATS LENGTH, LOTSIZE
A26   CALC L0,L1,L4,L5,C1
A27   LINEAR VS L0, L1
A28   ANALYSIS COMPANY, LENGTH, LOTS
A29   LINEAR VS L0, L1, L4
A30   REDI
A31   STATS L0, L1, L4
A32   ANAL PANEL COMP, LENGHT, LOTSI
A33   CALC L0,L1,L4,L5,L6,C1
A34   LINEAR VS L0,L1,L4,L5,L6
A35   STATS L0,L1,L4,L5,L6
A36   NEW WIDTH
A37   PLOT VS WIDTH
A38   NEW AREA
A39   PLOT VS AREA
A40   GROUP COUNT
A41   AN03R PANEL COMP, LENGTH, LOTS
A42   SELECT NULL
A43   CURVES*1 VS L0,L1,L4,L5,L6
A44   DELETE DV
A45   PLOT VS NULL
A46   NEW NULL
A47   SAMPLE X-Y PLOT
A48   START 1
A49   NEW LINE 1
A50   SELECT DATA
A51   MCDG SAMPLE PLOT VS LENGTH
A52   EXTRAP COST
A53   SAMPLE USAGE
A54   COEF OUTPUT
A55   COMPARISON BAR CHARTS
A56   SELECT DRIVER AND GRAPH
A57   SELECT LINEAL METHOD
A58   SELECT LINEAL MATERIAL
A59   SELECT LINEAL LOT SIZE
A60   SELECT LINEAL LENGTH
A61   IDENTIFY

  ******  PROFIL1  ******
ID        .NAME.
B1    CHOOSE
B2    XXX
B3    CALC
B4    TABLE
B5    COSTS
B6    PLOT
```

FIGURE 18. LIBRARY OF PROFILES ILLUSTRATED, USING PROFILE SHOW COMMAND

(1)
```
LOGIN,CLAYDON
ENTER BATTELLE PROJECT NUMBER FOR THIS SESSION
```

(2)
```
COMMAND- /LOGON,BASIS,MCDG
         MOUNTED BASISMP AND ISSMP
SAVEPFL=BASISPROFLCLAYDON
PROFIL1=BASISPROFLMCDG


              B A S I S

THE MANUFACTURE COST/DESIGN GUIDE SYSTEM

LAST UPDATE   06/30/78

TOTAL ITEMS IN DATA BASE=      6460

ENTER YOUR REQUESTS ONE AT A TIME
  1/
 PF CYCLE NO. = 018
 PF CYCLE NO. = 003
 PF CYCLE NO. = 001
```

(3)
```
PROFILE EXECUTE COMPARISON BAR CHARTS
ENTER THE PART TYPE : LINEAL OR PANEL
```

(4)
```
PICK ONE OF THE FOLLOWING..
..METHOD (MANUFACTURING METHOD)
..MATERIAL
..LOT SIZE
..LENGTH
ENTER THE COST DRIVER FOR COMPARISON : ONE OF THE ABOVE
```

(5)
```
PICK FROM THE FOLLOWING..
..BPCOST (BASE PART COST)
..JOGGLES (JOGGLES COST)
..FLHOLES (FLANGED HOLES COST)
..BEADS (BEADS COST)
..HTREAT (HEAT TREATMENT COST)
..SURFIN (SURFACE FINISH COST)
..TOLERAN (TOLERANCE COST)
..LINTRIM (LINEAL TRIM COST)
..ENDTRIM (END TRIM COST)
..UNFLHOLE (CUT-OUTS WITHOUT FLANGES COST)
ENTER THE COST ELEMENT(S) FROM THE LIST ABOVE.
COMBINATIONS SUCH AS BPCOST+JOGGLES+LINTRIM
CAN BE USED FOR COMPARING DISCRETE PART COSTS

ENTER A NUMBER, 1 TO 6, FOR BAR GRAPH TEXTURE
```

```
THE FOLLOWING ARE PARAMETERS TO BE SATISFIED
           PART TYPE ? LINEAL
              DRIVER ? METHOD
           COST ELEM ? BPCOST+JOGGLES
             TEXTURE ? 6

THE FOLLOWING ARE PARAMETERS TO BE SATISFIED
           PART CODE ? 2B
            MATERIAL ? ALUMINUM
              LENGTH ? 24
```

(6)
```
  6/ LIST


  .ITEMS. LINE      REQUEST
*   1148   1/  ACC:GT 400000000
*    128   2/  PARTCODE:2B
*     48   3/  MATERIAL:ALUMINUM
*     12   4/  LOTSIZE=25
*      3   5/  LENGTH:24
ENTER YOUR REQUEST
```

(7)

BAR NR.	PART CODE	MANUFACTURING METHOD	PART MATERIAL	LOT SIZE	LENGTH	WIDTH
1	2B	BRAKE + BUFFALO ROLL	ALUMINUM-2024	25	24.00	- - -
2	2B	BRAKE AND STRETCH	ALUMINUM-2024	25	24.00	- - -
3	2B	RUBBER PRESS	ALUMINUM-2024	25	24.00	- - -

FIGURE 19. MC/DG USAGE SAMPLE, SESSION 1

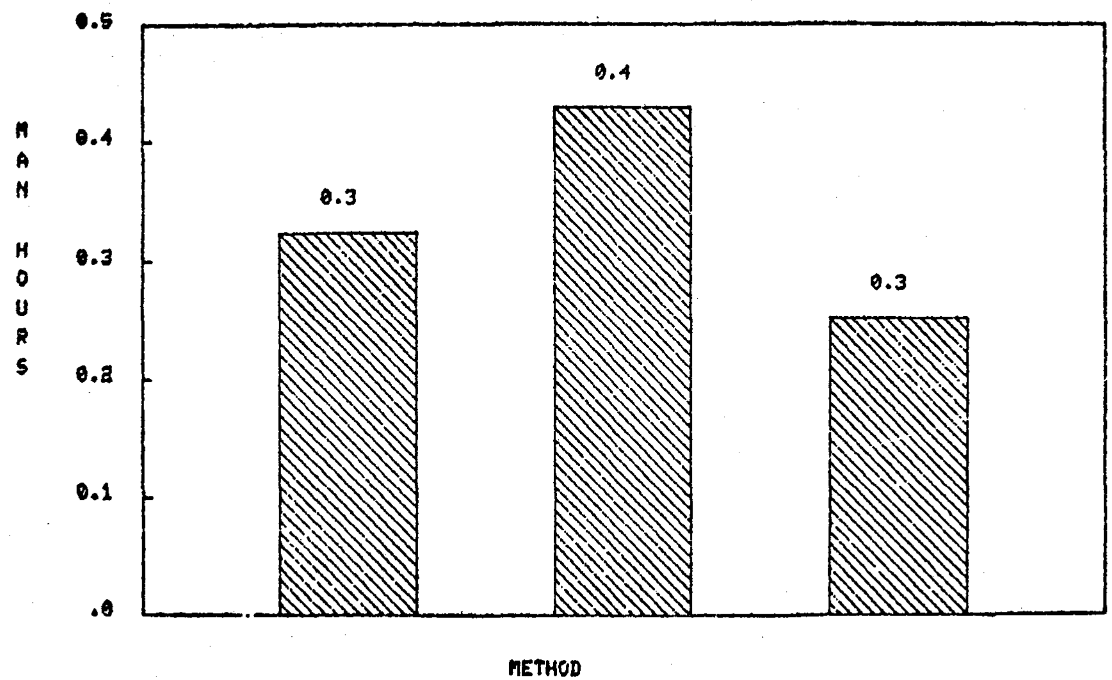


FIGURE 20. BAR CHART OF MAN-HOUR COST VERSUS MANUFACTURING METHODS

```
(1)  /LOGON,BASIS,MCDG
             MOUNTED BASISMP AND ISSMP
     SAVEPFL•BASISPPOFLCLAYDON
     PPGFIL1•BASISPPOFLMCDG


                  B A S I S

     THE MANUFACTURE COST/DESIGN GUIDE SYSTEM

     LAST UPDATE   06/30/78

     TOTAL ITEMS IN DATA BASE•      6460

     ENTER YOUR REQUESTS ONE AT A TIME
       1/


(2)  PROFILE EXECUTE SAMPLE USAGE
       1/ /X START 1

(3)  THE FOLLOWING ARE PARAMETERS TO BE SATISFIED
               GT/LT 400M ? GT
       2/ /X SELECT DATA
       2/ MATERIAL:*

       .ITEMS.     TERM
      IN YOUR DATA SUBSET
     A      536  MATERIAL:ALUMINUM
     B      536  MATERIAL:ALUMINUM-2024
     C      244  MATERIAL:PH15
     D      244  MATERIAL:STEEL
     E      244  MATERIAL:STEEL - PH15 - 7M0
     F      368  MATERIAL:TITANIUM
     G      368  MATERIAL:TITANIUM - 6AL - 4V
     H      536  MATERIAL:2024
     I      368  MATERIAL:4V
     J      368  MATERIAL:6AL

     K      244  MATERIAL:7M0
     END OF TERMS WITH YOUR STEM
     ENTER LETTERS TO BE COMBINED,
     SEPARATED BY COMMAS, OR ALL
        / D
             244 ITEMS
      IN YOUR DATA SUBSET
     SELECT TERMS, OR ENTER YOUR REQUEST
       3/ PARTCODE:*

       .ITEMS.     TERM
      IN YOUR DATA SUBSET
     A       32  PARTCODE:1A
     B       32  PARTCODE:1B
     C       32  PARTCODE:2A
     D       32  PARTCODE:2B
     E       32  PARTCODE:3A
     F       32  PARTCODE:3B
     G       36  PARTCODE:4
     H       16  PARTCODE:5
     END OF TERMS WITH YOUR STEM
     ENTER LETTERS TO BE COMBINED,
     SEPARATED BY COMMAS, OR ALL
        / C
              32 ITEMS
      IN YOUR DATA SUBSET
     SELECT TERMS, OR ENTER YOUR REQUEST
       4/ MFGMETH:*

       .ITEMS.     TERM
      IN YOUR DATA SUBSET
     A       16  MFGMETH:BRAKE
     B       16  MFGMETH:BRAKE FORM-COLD
     C       16  MFGMETH:COLD
     D       16  MFGMETH:FORM
     E       16  MFGMETH:PRESS
     F       16  MFGMETH:RUBBER
     G       16  MFGMETH:RUBBER PRESS
     END OF TERMS WITH YOUR STEM
     ENTER LETTERS TO BE COMBINED,
     SEPARATED BY COMMAS, OR ALL
        / B
              16 ITEMS
      IN YOUR DATA SUBSET
(4)  SELECT TERMS, OR ENTER YOUR REQUEST
       5/ SORT(LENGTH)


     BEGIN BASIS SORT
       5/ MFGMETH:BRAKE FORM-COLD **SORTED**
              16 ITEMS
(5)  ENTER YOUR REQUEST
       6/ /X CALC L0,L1,L4,L5,L6,C1
```

FIGURE 21. MC/DG USAGE SAMPLE, SESSION 2 (3 pages)

(6) ENTER YOUR REQUEST
?: PROFILE EXECUTE (OUTPUT-OFF) IDENTIFY

NR.	PART CODE	MANUFACTURING METHOD	PART MATERIAL	LOT SIZE	LENGTH	WIDTH
1	2A	BRAKE FORM-COLD	STEEL - PH15 - 7M0	5	24.00	- - -
2	2A	BRAKE FORM-COLD	STEEL - PH15 - 7M0	10	24.00	- - -
3	2A	BRAKE FORM-COLD	STEEL - PH15 - 7M0	25	24.00	- - -
4	2A	BRAKE FORM-COLD	STEEL - PH15 - 7M0	50	24.00	- - -
5	2A	BRAKE FORM-COLD	STEEL - PH15 - 7M0	5	48.00	- - -
6	2A	BRAKE FORM-COLD	STEEL - PH15 - 7M0	10	48.00	- - -
7	2A	BRAKE FORM-COLD	STEEL - PH15 - 7M0	25	48.00	- - -
8	2A	BRAKE FORM-COLD	STEEL - PH15 - 7M0	50	48.00	- - -
9	2A	BRAKE FORM-COLD	STEEL - PH15 - 7M0	5	96.00	- - -
10	2A	BRAKE FORM-COLD	STEEL - PH15 - 7M0	10	96.00	- - -
11	2A	BRAKE FORM-COLD	STEEL - PH15 - 7M0	25	96.00	- - -
12	2A	BRAKE FORM-COLD	STEEL - PH15 - 7M0	50	96.00	- - -
13	2A	BRAKE FORM-COLD	STEEL - PH15 - 7M0	5	144.00	- - -
14	2A	BRAKE FORM-COLD	STEEL - PH15 - 7M0	10	144.00	- - -
15	2A	BRAKE FORM-COLD	STEEL - PH15 - 7M0	25	144.00	- - -
16	2A	BRAKE FORM-COLD	STEEL - PH15 - 7M0	50	144.00	- - -

FIGURE 21. (Continued)

```
ENTEF YOUP PEQUEST
  6/ PUN CER(LPLOT)

THE FOLLOWING PEQUIRED PARAMETERS(S) NEED TO BE ENTERED
Y          X
   / X=L,Y=C1,XLABEL(PART LENGTH, IN.),YLABEL(MAN HOURS),+
CONTINUE
   / FRAME=3,CHAR(*)
```

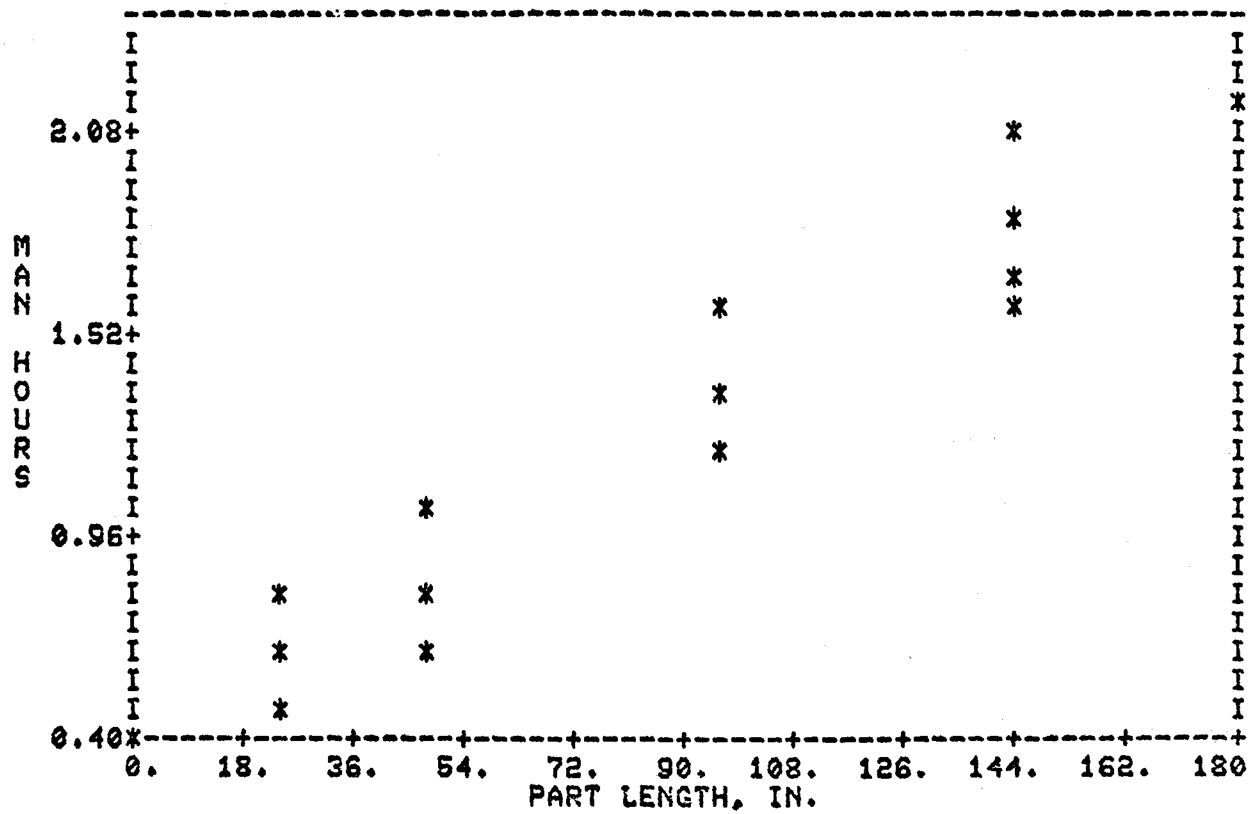
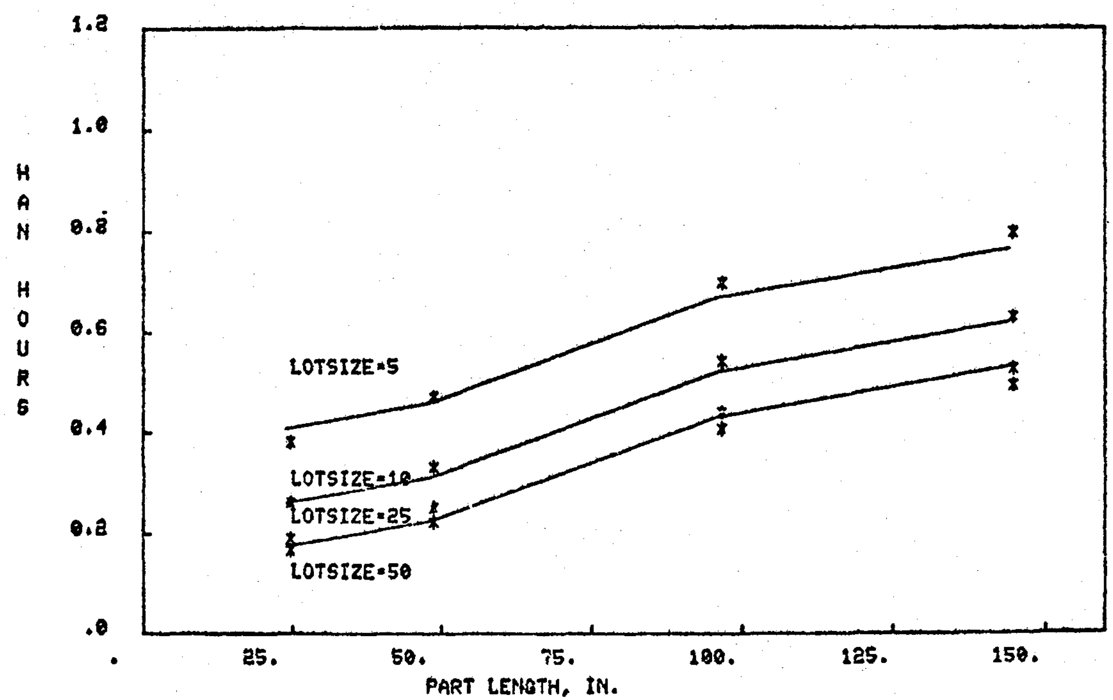


FIGURE 21. (Continued)



SAMPLE MC/DG PLOT SHOWING SEVERAL CURVES

ENTER P L O T COMMAND

/

STOP

FIGURE 22. MC/DG USAGE SAMPLE, SESSION 3 (3 PAGES)

```
PROFILE EXECUTE SAMPLE USAGE
  6  /% START 1


THE FOLLOWING ARE PARAMETERS TO BE SATISFIED
          GT/LT 400M ? GT
  7/ /% SELECT DATA
  7/ MATERIAL:*


  .ITEMS.    TERM
 IN YOUR DATA SUBSET
A      536  MATERIAL:ALUMINUM
B      536  MATERIAL:ALUMINUM-2024
C      244  MATERIAL:PH15
D      244  MATERIAL:STEEL
E      244  MATERIAL:STEEL - PH15 - 7M0
F      362  MATERIAL:TITANIUM
G      362  MATERIAL:TITANIUM - 6AL - 4V
H      536  MATERIAL:2024
I      368  MATERIAL:4V
J      368  MATERIAL:6AL
K      244  MATERIAL:7M0
END OF TERMS WITH YOUR STEM
ENTER LETTERS TO BE COMBINED,
SEPARATED BY COMMAS, OR ALL
   / A
       536 ITEMS
 IN YOUR DATA SUBSET
SELECT TERMS, OR ENTER YOUR REQUEST
  8/ PARTCODE:*


  .ITEMS.    TERM
 IN YOUR DATA SUBSET
A       32  PARTCODE:1A
B       48  PARTCODE:1B
C       16  PARTCODE:10
D       24  PARTCODE:12
E       32  PARTCODE:2A
F       48  PARTCODE:2B
G       32  PARTCODE:3A
H       42  PARTCODE:3B
I       32  PARTCODE:3C
J       16  PARTCODE:4A
K       32  PARTCODE:4B
L       32  PARTCODE:5A
M       32  PARTCODE:5B
N       32  PARTCODE:7A
O       32  PARTCODE:7B
P       16  PARTCODE:8
Q       32  PARTCODE:9
END OF TERMS WITH YOUR STEM
ENTER LETTERS TO BE COMBINED,
SEPARATED BY COMMAS, OR ALL
   / A
       32 ITEMS
 IN YOUR DATA SUBSET
SELECT TERMS, OR ENTER YOUR REQUEST
  9/ MFGMETH:*


  .ITEMS.    TERM
 IN YOUR DATA SUBSET
A       16  MFGMETH:BRAKE
B       16  MFGMETH:BRAKE FORM
C       16  MFGMETH:FORM
D       16  MFGMETH:PRESS
E       16  MFGMETH:RUBBER
F       16  MFGMETH:RUBBER PRESS
END OF TERMS WITH YOUR STEM
ENTER LETTERS TO BE COMBINED,
SEPARATED BY COMMAS, OR ALL
   / A

       16 ITEMS
 IN YOUR DATA SUBSET
SELECT TERMS, OR ENTER YOUR REQUEST
 10/ SORT(LENGTH)


BEGIN BASIS SORT
 10/ MFGMETH:BRAKE **SORTED**
       16 ITEMS
ENTER YOUR REQUEST
 11/ /% CALC L0,L1,L4,L5,L6,C1


THE FOLLOWING ARE PARAMETERS TO BE SATISFIED
          COST ELEM ? BPCOST
```

FIGURE 22. (Continued)

```
ENTER YOUR REQUEST
 11/ /X IDENTIFY
 11/ FORMAT(WRAP,LL=65)
ENTER FORMAT SPECIFICATION
   / '    PART MANUFACTURING          PART       LOT  ',/,+
   / 'NR. CODE    METHOD         MATERIAL    SIZE LENGTH  WIDTH',/
   / '+
   / '--- ---- ------------- -------------- ---- ------ ------',/
   / +
     (NR=I3,1X,PARTCODE=A4,1X,MFGMETH=C13,1X,MATERIAL=C14,1X,LOTS
     IZE=I4,1X,LENGTH=F6.2,1X,WIDTH=F6.2)

    PART MANUFACTURING       PART       LOT
NR. CODE    METHOD         MATERIAL    SIZE LENGTH  WIDTH
--- ---- ------------- -------------- ---- ------ ------
  1 1A   BRAKE FORM    ALUMINUM-2024     5  24.00 - - -
  2 1A   BRAKE FORM    ALUMINUM-2024    10  24.00 - - -
  3 1A   BRAKE FORM    ALUMINUM-2024    25  24.00 - - -
  4 1A   BRAKE FORM    ALUMINUM-2024    50  24.00 - - -
  5 1A   BRAKE FORM    ALUMINUM-2024     5  42.00 - - -
  6 1A   BRAKE FORM    ALUMINUM-2024    10  48.00 - - -
  7 1A   BRAKE FORM    ALUMINUM-2024    25  48.00 - - -
  8 1A   BRAKE FORM    ALUMINUM-2024    50  48.00 - - -
  9 1A   BRAKE FORM    ALUMINUM-2024     5  96.00 - - -
 10 1A   BRAKE FORM    ALUMINUM-2024    10  96.00 - - -
 11 1A   BRAKE FORM    ALUMINUM-2024    25  96.00 - - -
 12 1A   BRAKE FORM    ALUMINUM-2024    50  96.00 - - -
 13 1A   BRAKE FORM    ALUMINUM-2024     5 144.00 - - -
 14 1A   BRAKE FORM    ALUMINUM-2024    10 144.00 - - -
 15 1A   BRAKE FORM    ALUMINUM-2024    25 144.00 - - -
 16 1A   BRAKE FORM    ALUMINUM-2024    50 144.00 - - -

RUN PLAT()
ENTER P L O T COMMAND
   . PLOTXY,X=L,Y1=C1,Y2=COST1,Y3=COST2,Y4=COST3,Y5=COST4,
ENTER P L O T COMMAND
   / XTITLE(PART LENGTH, IN.),YTITLE(MAN HOURS),
ENTER P L O T COMMAND
   / PLOTTITLE(SAMPLE MC/DG PLOT SHOWING SEVERAL CURVES),
ENTER P L O T COMMAND
   . XMIN=0.,XMAX=160.,YMIN=0.,YMAX=1.2,
ENTER P L O T COMMAND
   / CHAR(*,LINE,LINE,LINE,LINE),END
```

FIGURE 22. (Continued)

```
423682201   290GROUP TECH
423682201   390MC-DG DATA
423682201   490AVGERAGES
423682201   590IIA- AL - 3
423682201   690ALUMINUM-2024
423682201   790
423682201   890ASSEMBLY METHOD = MANUAL
423682201   990 11/07/78
423682201 1090THIS IS THE STATISTICS RECORD
423682201 1190
423682201 1290MECHANICALLY FASTENED ASSEMBLY; DOOR PANEL
423682201 1390MANUAL;    707
423682201 1490
423682201 1590
423682201 1690      24.000
423682201 1790      72.000
423682201 1890    1728.000
423682201 1990       0.000
423682201 2090       0.000
423682201 2190       0.000
423682201 2290       0.000
423682201 2390      68.000
423682201 2490     707.000
423682201 2590      17.437
423682201 2690       8.630
423682201 2790       0.000
423682201 2890       0.000
423682201 2990       0.000
423682201 3090       0.000
423682201 3190     612.333
423682201 3290       0.000
423682201 3390       0.000
423682201 3490       0.000
423682201 3590       0.000
423682201 3690       0.000
423682201 3790       0.000
423682201 3890       0.000
423682201 3990       0.000
423682201 4090       0.000
423682201 4190       0.000
423682201 4290       0.000
423682201 4390       0.000
423682201 4490       0.000
423682201 4590       0.000
423682201 4690     474.000
423682201 4790     185.000
423682201 4890      32.000
423682201 4990      16.000
423682201 5090       0.000
```

FIGURE 23. PHASE II PREPROCESSED MECHANICAL ASSEMBLY DATA RECORD

```
422122201  290GROUP TECH
422122201  390MC-DG DATA
422122201  490AVGERAGES
422122201  590IIA- AL - 2
422122201  690ALUMINUM-2024
422122201  790
422122201  890ASSEMBLY METHOD = MANUAL
422122201  990 10/19/78
422122201 1090THIS IS THE STATISTICS RECORD
422122201 1190
422122201 1290MECHANICALLY FASTENED ASSEMBLY; DOOR PANEL
422122201 1390MANUAL;    298
422122201 1490
422122201 1590
422122201 1690     24.000
422122201 1790     72.000
422122201 1890   1728.000
422122201 1990      0.000
422122201 2090      0.000
422122201 2190      0.000
422122201 2290      0.000
422122201 2390     12.000
422122201 2490    298.000
422122201 2590      7.273
422122201 2690      4.900
422122201 2790      0.000
422122201 2890      0.000
422122201 2990      0.000
422122201 3090      0.000
422122201 3190    512.667
422122201 3290      0.000
422122201 3390      0.000
422122201 3490      0.000
422122201 3590      0.000
422122201 3690      0.000
422122201 3790      0.000
422122201 3890      0.000
422122201 3990      0.000
422122201 4090      0.000
422122201 4190      0.000
422122201 4290      0.000
422122201 4390      0.000
422122201 4490      0.000
422122201 4590      0.000
422122201 4690    279.000
422122201 4790      8.000
422122201 4890     11.000
422122201 4990      0.000
422122201 5090      0.000
```

FIGURE 23. (Continued)

```
834020104   190834020104
834020104   290GROUP TECH
834020104   390MC-DG DATA
834020104   490AVGERAGES
834020104   590IIB - C - 4
834020104   690AS/3501-6 RESIN CONTENT 45+-3%
834020104   790
834020104   890HAND LAYUP/AUTOCLAVE ADHESIVE-BONDED ASSEMBLY
834020104   990 11/08/78
834020104 1090THIS IS THE STATISTICS RECORD
834020104 1190
834020104 1290COMPOSITE PART, SKIN PANEL SINGLE CURVATURE;
834020104 1390
834020104 1490
834020104 1590
834020104 1690     48.000
834020104 1790     48.000
834020104 1890   2304.000
834020104 1990      0.000
834020104 2090      0.000
834020104 2190      0.000
834020104 2290      0.000
834020104 2390      0.000
834020104 2490      0.000
834020104 2590      0.000
834020104 2690      6.753
834020104 2790      0.000
834020104 2890      0.000
834020104 2990      0.000
834020104 3090      0.000
834020104 3190    283.333
834020104 3290      0.000
834020104 3390      0.000
834020104 3490      0.000
834020104 3590      0.000
834020104 3690      0.000
834020104 3790      0.000
834020104 3890      0.000
834020104 3990      0.000
834020104 4090      0.000
834020104 4190      0.000
834020104 4290      0.000
834020104 4390      0.000
834020104 4490      0.000
834020104 4590      0.000
83402010413090          2
83402010413190          2
83402010413290          2
83402010413390          0
83402010413490         16
83402010413590          0
```

FIGURE 24. PHASE II PREPROCESSED COMPOSITE PART DATA RECORD

```
831164206   190831164206
831164206   290GROUP TECH
831164206   390MC-DG DATA
831164206   490AVGERAGES
831164206   590IIB - C - 1
831164206   690AS/3501-6 RESIN CONTENT 45+-3%
831164206   790
831164206   890HAND LAYUP/AUTOCLAVE CURING COMPONENTS
831164206   990 11/08/78
831164206 1090THIS IS THE STATISTICS RECORD
831164206 1190
831164206 1290COMPOSITE PART, STRAIGHT HAT CONSTANT SECTION;
831164206 1390
831164206 1490
831164206 1590
831164206 1690     96.000
831164206 1790      6.000
831164206 1890    576.000
831164206 1990      0.000
831164206 2090      0.000
831164206 2190      0.000
831164206 2290      0.000
831164206 2390      0.000
831164206 2490      0.000
831164206 2590      0.000
831164206 2690     12.763
831164206 2790      0.000
831164206 2890      0.000
831164206 2990      0.000
831164205 3090      0.000
831164206 3190    240.000
831164206 3290      0.000
831164206 3390      0.000
831164206 3490      0.000
831164206 3590      0.000
831164206 3690      0.000
831164206 3790      0.000
831164206 3890      0.000
831164206 3990      0.000
831164206 4090      0.000
831164206 4190      0.000
831164206 4290      0.000
831164206 4390      0.000
831164206 4490      0.000
831164206 4590      0.000
83116420613090          16
83116420613190           8
83116420613290           8
83116420613390           4
83116420613490           0
83116420613590           0
```

FIGURE 24. (Continued)

```
832164204  190832164204
832164204  290GROUP TECH
832164204  390MC-DG DATA
832164204  490AVGERAGES
832164204  590IIB - C - 2
832164204  690AS/3501-6 RESIN CONTENT 45+-3%
832164204  790
832164204  890HAND LAYUP/AUTOCLAVE CURING COMPONENTS
832164204  990 11/08/78
832164204 1090THIS IS THE STATISTICS RECORD
832164204 1190
832164204 1290COMPOSITE,PART, CURVED J CONSTANT SECTION;
832164204 1390
832164204 1490
832164204 1590
832164204 1690     96.000
832164204 1790      4.500
832164204 1890    432.000
832164204 1990      0.000
832164204 2090      0.000
832164204 2190      0.000
832164204 2290      0.000
832164204 2390      0.000
832164204 2490      0.000
832164204 2590      0.000
832164204 2690     19.833
832164204 2790      0.000
832164204 2890      0.000
832164204 2990      0.000
832164204 3090      0.000
832164204 3190    375.667
832164204 3290      0.000
832164204 3390      0.000
832164204 3490      0.000
832164204 3590      0.000
832164204 3690      0.000
832164204 3790      0.000
832164204 3890      0.000
832164204 3990      0.000
832164204 4090      0.000
832164204 4190      0.000
832164204 4290      0.000
832164204 4390      0.000
832164204 4490      0.000
832164204 4590      0.000
83216420413090         13
83216420413190          8
83216420413290          6
83216420413390          4
83216420413490          0
83216420413590          0
```

FIGURE 24. (Continued)

COMPOSITE I SECTION RECURRING COST/PART

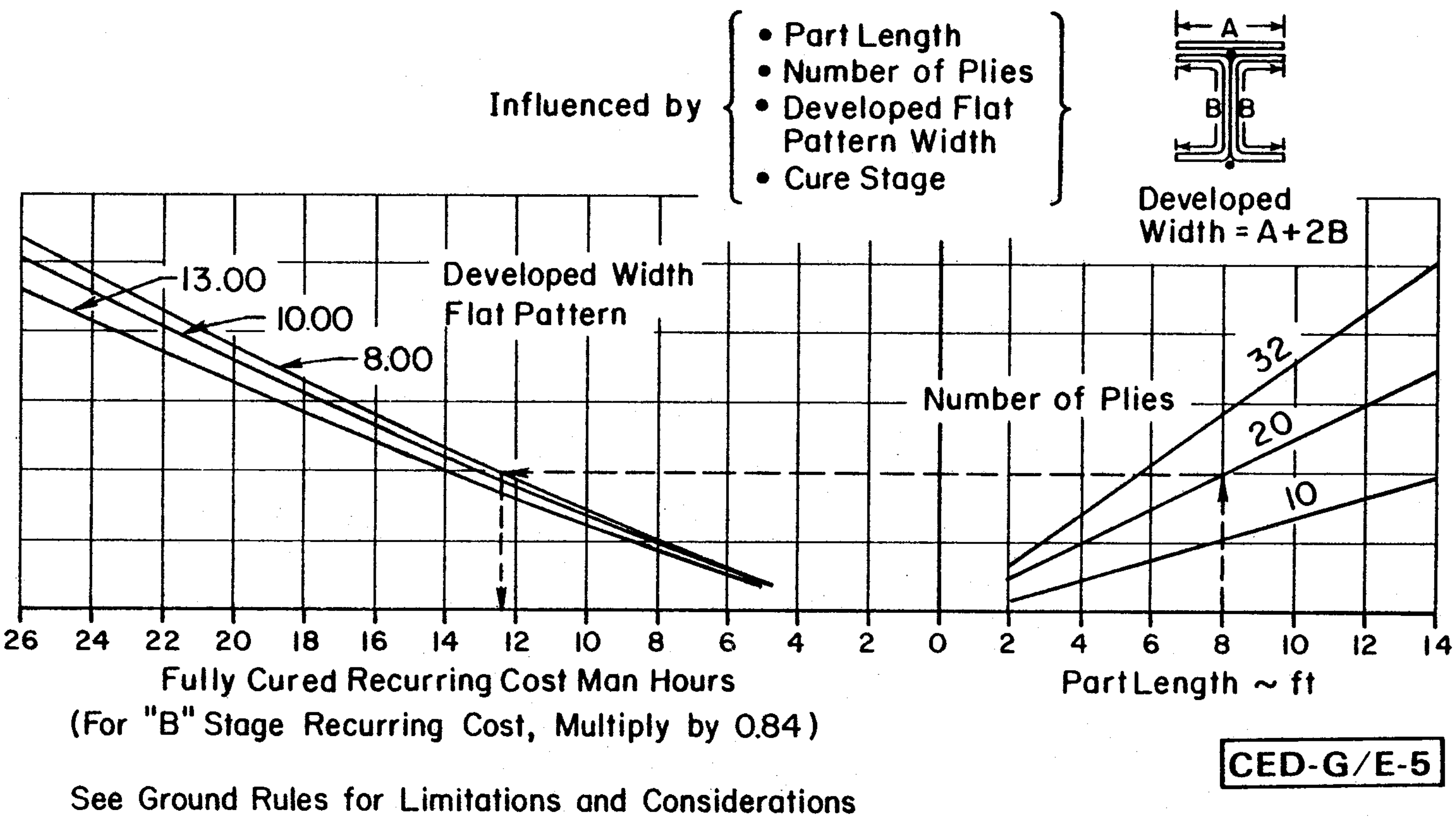


FIGURE 25. CED COMPOSITE I SECTION RECURRING COST/PART

```
PLEASE LOGIN
LOGIN,CLAYDON,     ,SUP,X
ENTER BATTELLE PROJECT NUMBER FOR THIS SESSION


   COMMAND- ETL,500
(1)COMMAND- /LOGON,BASIS,PHASEII
            *MOUNTED* BASISMP   ISSMP
   SAVEPFL=BASISPROFLPHASEII
   PROFIL1=BASISPROFLDEMO
   PROFIL2=BASISPROFLCLAYDON
   PROFIL3=BASISPROFLMCDG


                B A S I S

   THE MANUFACTURE COST/DESIGN GUIDE SYSTEM

   LAST UPDATE   11/22/78

   TOTAL ITEMS IN DATA BASE=      7283

   ENTER YOUR REQUESTS ONE AT A TIME
     1/
    PF CYCLE NO. = 040
    PF CYCLE NO. = 001


(2) PROFILE EXECUTE START COMPOSITE


   THE FOLLOWING ARE PARAMETERS TO BE SATISFIED
(3)            PART SHAPE ? I
(4)                 DRIVER ? LENGTH
      1/ SUBFILE:AVGERAGES
           1347 ITEMS
      2/ PARTFORM:COMPOSITE PART
              96 ITEMS
    IN YOUR DATA SUBSET
      3/ DESCRIBE:I
              29 ITEMS
    IN YOUR DATA SUBSET
      4/ /X GRAPH COMPOSITE V LENGTH
```

FIGURE 26. SAMPLE OF PHASE II MC/DG USAGE

```
   THE FOLLOWING ARE PARAMETERS TO BE SATISFIED
(5)           PLY COUNT ? ALL
(6)           PART WIDTH ? 3
(7)           PART SHAPE ? I
    4/ /X PLY ALL COMPOSITE
    4/ PLYO:*ALL
          29 ITEMS
    5/ /X WIDTH 3 COMPOSITE
    5/ WIDTH:3 TO 3.99
          9 ITEMS
   IN YOUR DATA SUBSET
    6/ /X COST COMPOSITE

   THE FOLLOWING ARE PARAMETERS TO BE SATISFIED
(8)           COST ELEM ? BPCOST
    6/ /X TABLE COMPOSITE
```

MC/DG DATA FOR ADVANCED COMPOSITE FABRICATION

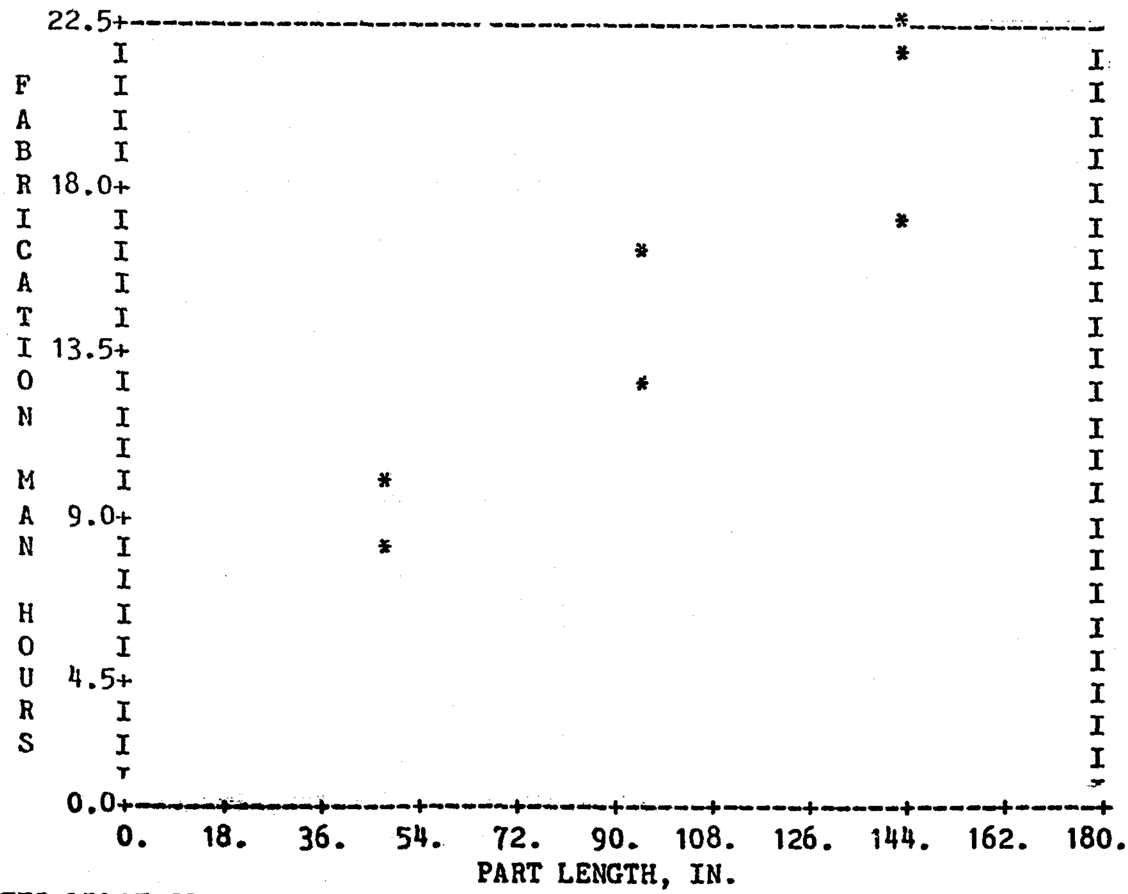
FLAT PATTERN				PLY COUNT				STRIP PLIES			BASE PART MAN HOURS	NONRECUR. TOOLING MAN HOURS
L	X	W	AREA	0	45	90	TOT	0	45	90	HOURS	MAN HOURS
48	X	3.50-	- -	8	8	4	20	4	0	0	7.95	268.3
96	X	3.50-	- -	8	8	4	20	4	0	0	12.50	359.3
144	X	3.50-	- -	8	8	4	20	4	0	0	16.89	476.7
48	X	3.50-	- -	16	8	8	32	4	0	0	9.58	268.3
96	X	3.50-	- -	16	8	8	32	4	0	0	15.97	359.3
144	X	3.50-	- -	16	8	8	32	4	0	0	21.82	476.7
48	X	3.50-	- -	16	16	0	32	8	0	0	9.84	268.3
96	X	3.50-	- -	16	16	0	32	8	0	0	16.41	359.3
144	X	3.50-	- -	16	16	0	32	8	0	0	22.43	476.7

```
    6/ RUN CER(LPLOT,FRAME=3,CHAR(*),XMIN=0,XMAX=180,YMIN=0)

   THE FOLLOWING REQUIRED PARAMETERS(S) NEED TO BE ENTERED
   Y          X
   CONTINUE
      / X=DRIVER,Y=COMPCOST,XLABEL(PART LENGTH, IN.),YLABEL(FABRICATION M
      / AN HOURS),PLOTLABEL(COMPOSITE PART COST, MAN HOURS)
```

FIGURE 26. (Continued)

COMPOSITE PART COST, MAN HOURS




```
ENTER LPLOT COMMAND
   / STOP
ENTER YOUR REQUEST
  6/ LOGOUT

MC/DG RETURNS YOU TO INTERCOM

CONNECT TIME 0 HRS.  7 MIN.

GOODBYE
```

FIGURE 26. (Continued)

```
               B A S I S
THE MANUFACTURE COST/DESIN GUIDE SYSTEM

LAST UPDATE   11/13/78

TOTAL ITEMS IN DATA BASE=      7328

ENTER YOUR REQUESTS ONE AT A TIME
  1/ PROFILE EXECUTE START ASSEMBLY


THE FOLLOWING ARE PARAMETERS TO BE SATISFIED
            MATERIAL ? TITANIUM
  1/ SUBFILE:AVGERAGES
      1357 ITEMS
  2/ PARTFORM:MECHANICALLY FASTENED ASSEMBLY
        74 ITEMS
 IN YOUR DATA SUBSET
  3/ MATERIAL:TITANIUM
        36 ITEMS
 IN YOUR DATA SUBSET
  4/ DESCRIBE:AVIONICS**3


  .ITEMS.    TERMS
 IN YOUR DATA SUBSET
A       12  DESCRIBE:AVIONICS
B       12  DESCRIBE:BAY
C       24  DESCRIBE:DOOR
ENTER  MORE  TO CONTINUE TERM LIST
ENTER LETTERS TO BE COMBINED,
SEPARATED BY COMMAS, OR ALL

   / C
        24 ITEMS
 IN YOUR DATA SUBSET
SELECT TERMS, OR ENTER YOUR REQUEST
  5/ INSTMETH:AUTO**4


  .ITEMS.    TERMS
 IN YOUR DATA SUBSET
A        8  INSTMETH:AUTO
B        8  INSTMETH:AUTOMATIC
C        8  INSTMETH:MAN
D        8  INSTMETH:MANUAL
ENTER  MORE  TO CONTINUE TERM LIST
ENTER LETTERS TO BE COMBINED,
SEPARATED BY COMMAS, OR ALL

   / B,D
        16 ITEMS
 IN YOUR DATA SUBSET
SELECT TERMS, OR ENTER YOUR REQUEST
```

FIGURE 27. SAMPLE OF USAGE FOR MECHANICALLY FASTENED ASSEMBLIES

MC/DG DATA FOR MECHANICALLY FASTENED ASSEMBLIES

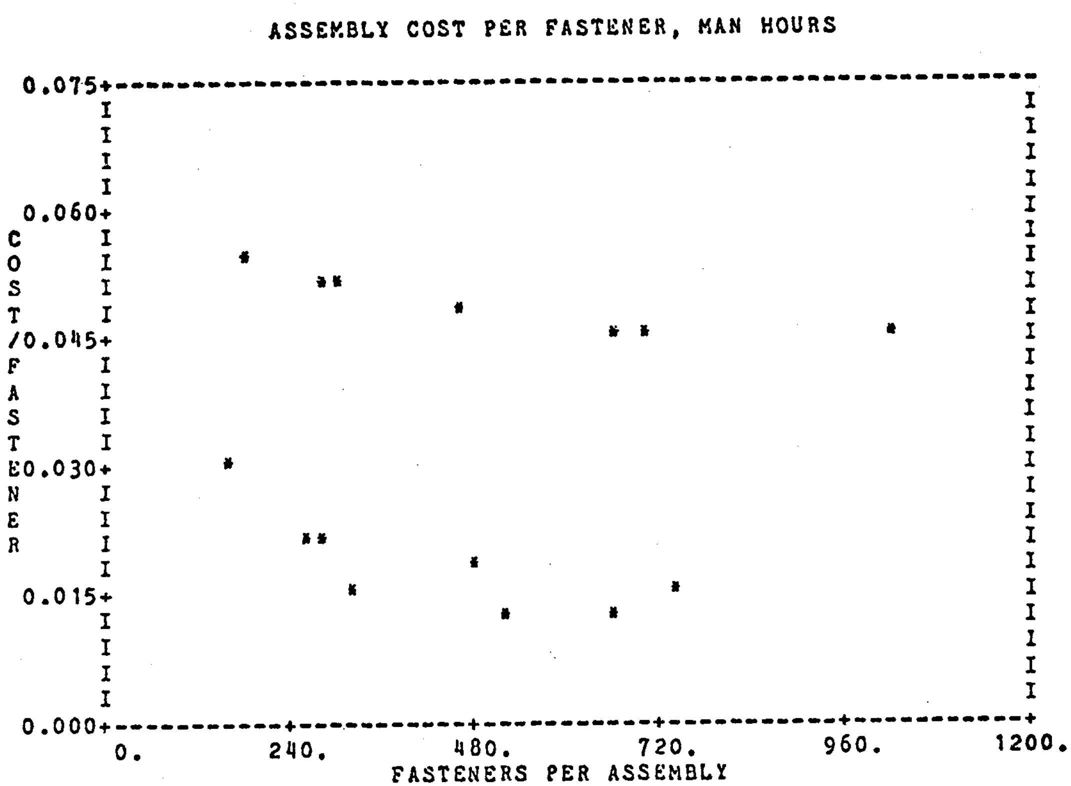
DESCRIPTION		FASTENER INSTALLATION METHOD			FASTENER TYPE				TOTAL FASTENER COUNT	BASE ASSEMB. MAN HOURS	TOTAL ASSEMB. MAN HOURS	NONRECUR TOOLING MAN HOURS
ASSEMBLY SIZE	NR OF PARTS	AUTO	MANUAL	COMB	1	2	3	4				
24 X 36	7	AUTOMATIC;156			159-	-	- -	- -	159	4.83	4.83	390.
24 X 72	10	AUTOMATIC;266			266-	-	- -	- -	266	5.70	5.70	633.
48 X 36	10	AUTOMATIC;276			258	8	10	- -	276	5.73	5.73	534.
24 X 36	9	AUTOMATIC;314			314-	-	- -	- -	314	4.67	4.67	515.
24 X 72	16	AUTOMATIC;474			474-	-	- -	- -	474	9.17	9.17	782.
48 X 36	11	AUTOMATIC;516			516-	-	- -	- -	516	6.54	6.54	659.
48 X 96	14	AUTOMATIC;664			626	8	30	- -	664	8.64	8.64	1034.
48 X 96	19	AUTOMATIC;736			736-	-	- -	- -	736	10.30	10.30	1204.
24 X 36	10	MANUAL; 175			162	8	5	- -	175	4.83	9.32	352.
48 X 36	12	MANUAL; 284			266	8	10	- -	284	5.73	14.07	458.
24 X 72	12	MANUAL; 298			279	8	11	- -	298	5.70	14.88	513.
24 X 36	43	MANUAL; 462			314	100	32	16	462	4.67	22.78	451.
48 X 96	16	MANUAL; 664			626	8	30	- -	664	8.64	29.92	816.
48 X 36	43	MANUAL; 664			516	100	32	16	664	6.54	29.68	547.
24 X 72	68	MANUAL; 707			474	185	32	16	707	9.17	32.21	610.
48 X 96	83	MANUAL; 1011			736	227	32	16	1011	10.30	46.03	923.

FIGURE 27. (Continued)

```
  7/ /X GRAPH ASSEMBLY
  7/ RUN CER(LPLOT,FRAME=3,CHAR(*),XMIN=0,XMAX=1000)

THE FOLLOWING REQUIRED PARAMETERS(S) NEED TO BE ENTERED
Y         X


   / X=FC,Y=UTAHR,XLABEL(FASTENERS PER ASSEMBLY),YLABEL(COST/FASTENER)
   / ,PLOTLABEL(ASSEMBLY COST PER FASTENER, MAN HOURS),YMIN=0.0,YMAX=0
   / .06
```




```
ENTER LPLOT COMMAND
   / STOP
ENTER YOUR REQUEST
  7/
```

FIGURE 27. (Continued)

SECTION VI

OPPORTUNITIES FOR FURTHER DEVELOPMENT OF THE COMPUTERIZED MC/DG

The construction and evaluation of the demonstration section of the computerized MC/DG has revealed several possible opportunities for further evolution of the system. The areas of opportunity, discussed in more detail later in this report, include making the system more "dynamic", incorporating a more adaptable graphics package, and developing an interactive training procedure for the system. A more "dynamic" system is one in which the designer could utilize the computer in an interactive mode to perform many of the tasks that would be time-consuming and bothersome if done by hand. Examples of this would be to determine the impact of labor rate and/or material price fluctuations, and to extrapolate or interpolate data in the data base. The new graphics package would provide more flexibility in format presentation and allow easier designer modification or creation of formats. The on-line training procedure would allow the potential user to learn the system at his own pace as fits his schedule.

1. APPLICATION OF DISSPLA TO THE COMPUTERIZED MC/DG

The Display Integrated Software System and Plotting Language (DISSPLA), developed by the Integrated Software System Corporation (ISSCO) and recently acquired by Battelle, has features which make it attractive for use with the computerized "Manufacturing Cost/Design Guide (MC/DG)". The features that make DISSPLA attractive for use with the MC/DG computerized system are:

- Portability
- Ease of use
- Ability to handle both graphic and textural modes of data presentation
- Color graphics.

The DISSPLA system could be used to produce the data presentation formats suggested in the MC/DG reports. Because of the ability of the system to handle conventional x-y plots, bar graphs, pie charts, and

three-dimensional representations, new formats may be developed in future MC/DG efforts to better utilize the capabilities of DISSPLA.

It should be noted that DISSPLA is not a graphics package designed to produce production quality engineering drawings. DISSPLA was specifically designed to present data in a graphical form. Because of this, simple plots can be produced with very few instructions, which could be helpful to a designer using the computerized MC/DG who wanted to display his data in a form not included as one of the standard formats.

a. Use of DISSPLA with MC/DG

At present, computer core space limitations prevent the full integration of DISSPLA with the computerized MC/DG. It may be possible to use the BASIS FORMAT report generator module to write the data to be plotted, including curve and axis labels, onto a permanent file. Then, using the capabilities of BASIS to execute a program exterior to BASIS, run DISSPLA utilizing the permanent data file created by the BASIS FORMAT module.

If this method of utilizing DISSPLA with BASIS is possible, some interesting MC/DG formats may be displayed for the designer using the computerized MC/DG.

Some other capabilities of DISSPLA are:

- Multiple axes
- Curve smoothing and curve fitting
  - Polynomial curve fits (third and fourth order)
  - Spline interpolation
- 3-D plotting
- Multiple plots per page.

Further investigation to determine the full potential of using DISSPLA with the computerized MC/DG should be considered, so that the computerized MC/DG can be the best computerized tool ever offered to industry.

2. TRAINING

The tutorials and training aids incorporated into the computerized MC/DG offer an opportunity for creative development. The inclusion of an on-line training system, similar to Control Data Corporation's

PLATO system, would allow designers to learn the use of the guide at their own pace and as their schedule would allow.

The PLATO computer-based education system does not appear to be directly applicable to the MC/DG because of hardware and software problems, but could serve as an excellent guide for the design of the on-line MC/DG tutorials and training aids.

3. DESIGN USES FOR A "DYNAMIC" COMPUTERIZED MC/DG

The computerized MC/DG can be utilized by a designer to perform many tasks to determine often critical information that would be time-consuming, intricate, and bothersome if he had to do the tasks by hand. Several of these tasks are described below.

One possible use of a "dynamic" computerized MC/DG would be to determine the impact of material price fluctuations. With inflation and advanced material production methods both contributing to change the cost of materials, the ability to use current and projected material costs is a vital need in all phases of design. This is especially true of conceptual and preliminary designers attempting to incorporate a greater percentage of composite materials into future aircraft. These designers are faced with constantly changing material costs, influenced by increasing use of the materials, and new methods of producing the fibers. These factors can cause a trade study to become obsolete almost over night. Without a dynamic computerized MC/DG, the number of trade studies performed would be severely limited and a more nearly optimum application of composite materials would not be possible.

Labor rate fluctuations could be handled in much the same way as the material price variations. As labor rates grow progressively higher, the need to design a part that can be manufactured with the least amount of hands on labor will become more important. With the computerized MC/DG, the designer could use projected labor rate values for the proposed time period of production in his trade study, to determine if the labor rate would cause a major problem in the cost of the project. Figure 28 shows a proposed format that could be used to display the effect of the labor rate (or material price) fluctuations.

Determination of the influence of aircraft buy quantity on the location on the learning curve can be easily included in the trade

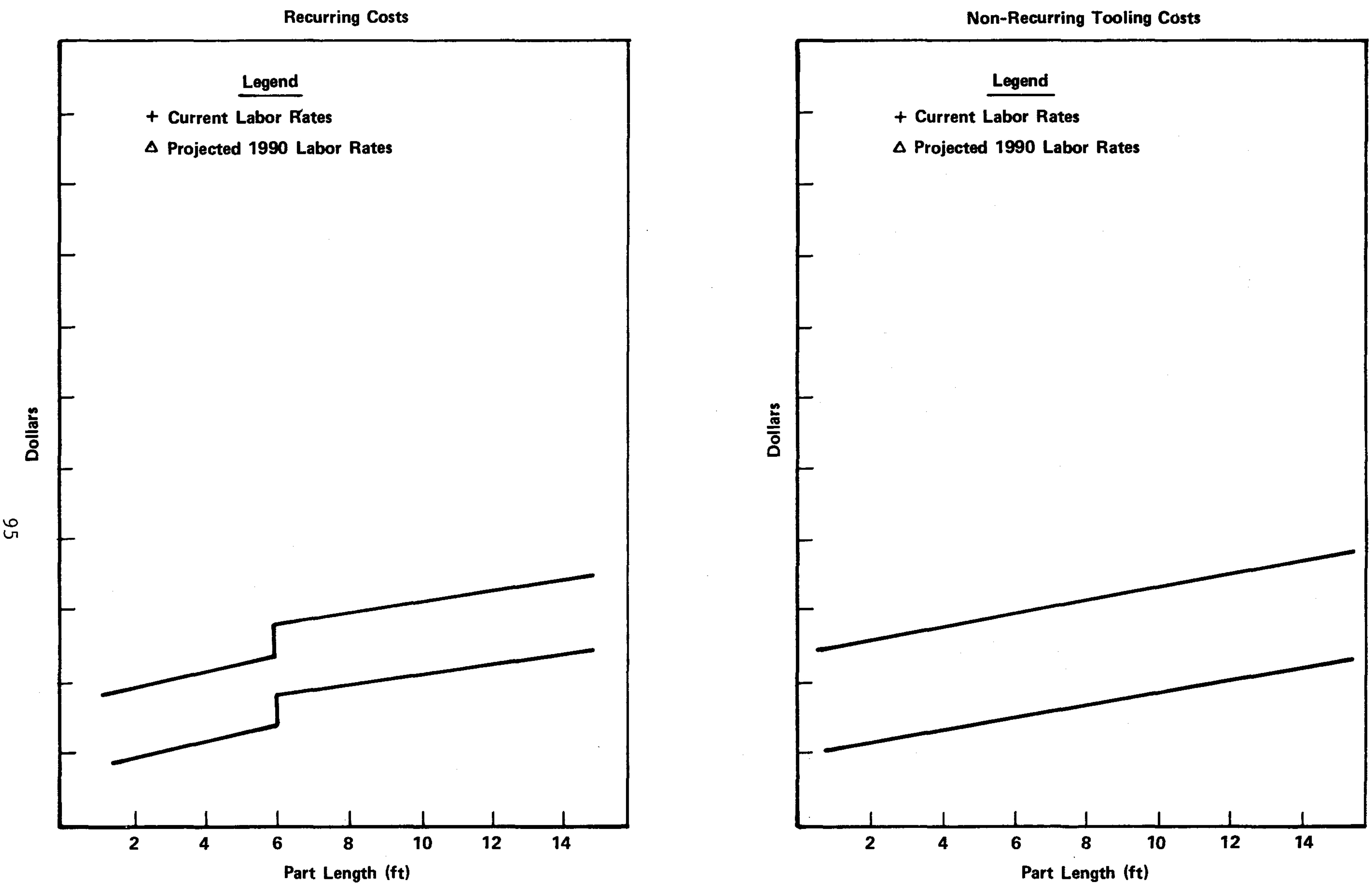


FIGURE 28. CED ALUMINUM STRAIGHT ANGLE BRAKE FORM

studies utilizing the computerized MC/DG. The current MC/DG data are based on a unit 200 learning curve, but a prototype development of maybe five aircraft would have a much higher value on the learning curve. On the other end of the scale, a very large production contract would have a much lower value on the learning curve. The impact of this learning curve value could be a major factor in management decisions to determine if a bid should be presented for a potential contract. With a computerized MC/DG, the designer could quickly determine the point at which it would be practical to submit a bid (given a target by management).

A "dynamic" computerized MC/DG would also be of use in determining the impact of lot release size, especially for lot sizes of less than 25 units. Beyond 25 units, the impact of lot size is negligible for trade-study purposes, but as the lot release size decreases below 25 units, the impact of lot size increases dramatically. With a computerized MC/DG, the designer, in cooperation with production planning personnel and management, could perform trade studies to determine an optimum design for various lot release sizes. Examples of proposed formats to illustrate the impact of lot release size are shown in Figures 29 and 30.

The computer would be an invaluable aid in extrapolating and interpolating dimensional data of parts and assemblies. This function of the computerized MC/DG is, in reality, more of a necessity than a convenience, because the data base could not contain all possible combinations of dimensions for aerospace parts. In order to conduct a trade study, the designer must be able to input the part dimensions and have the computer return the desired data.

Another helpful feature of a computerized MC/DG would be the ability to retrieve earlier design trade-off data in a readily usable and recognizable form. This would allow the designer to quickly evaluate past designs and determine what features would be applicable to his particular problem and what to avoid. This retrieval feature would also be helpful to designers in preparing presentations to management detailing how the chosen part configuration was developed, thus providing both the designer and management with confidence that the best possible part configuration had been chosen, within the constraints provided.

There are surely more possible design uses of a "dynamic" MC/DG than have been presented in this brief discussion, but the above examples show that to be a successful design tool, the computerized MC/DG must be a "dynamic" rather than a "static" system.

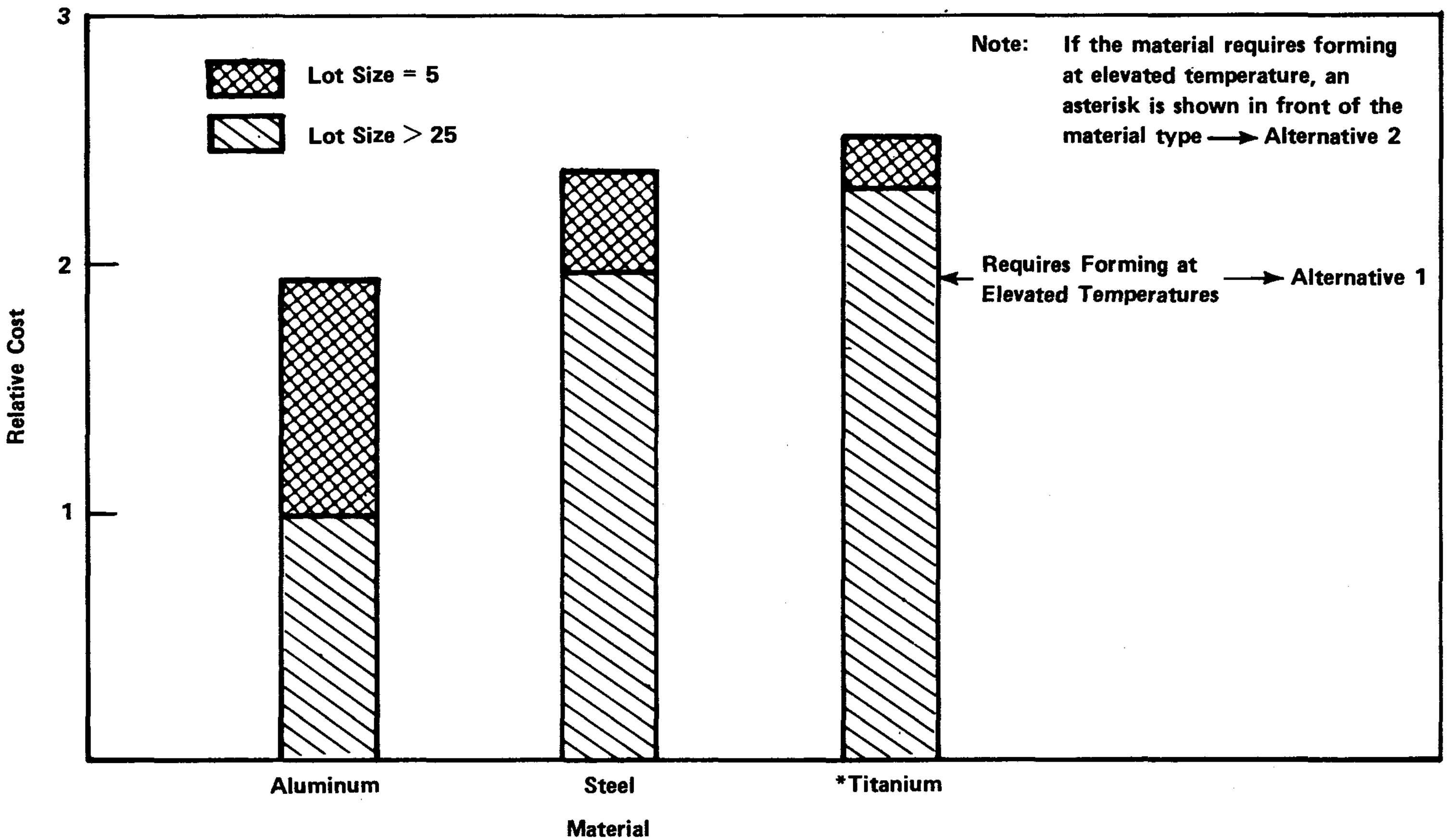


FIGURE 29. CED COST EFFECTS OF MATERIAL AND LOT SIZE

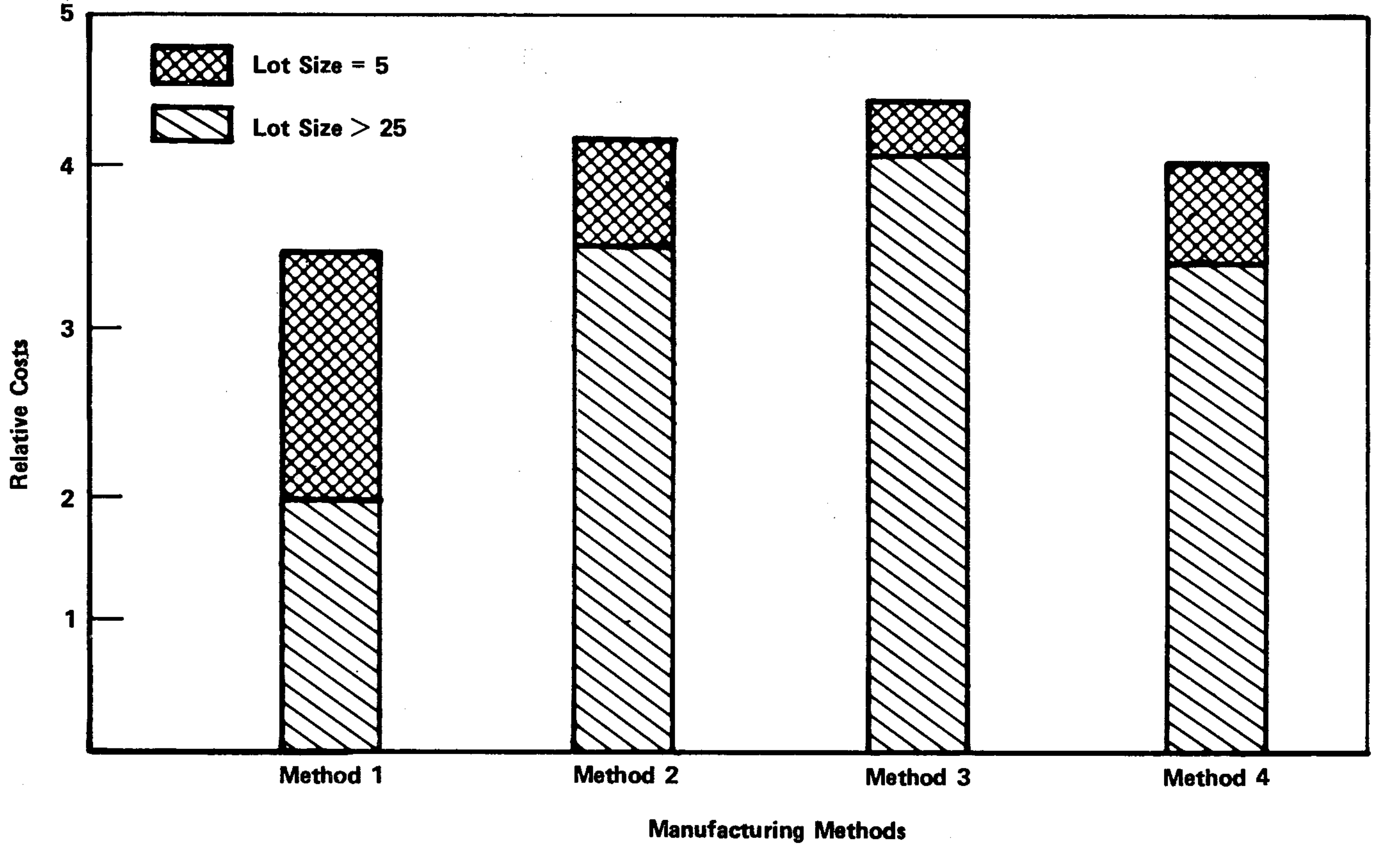


FIGURE 30. CED COST EFFECTS OF MANUFACTURING METHODS AND LOT SIZE FOR MATERIAL PART SHAPE

SECTION VII

IMPLEMENTATION PLAN FOR THE COMPUTERIZED MC/DG

1. INTRODUCTION

The purpose of this plan is to outline how a full-scale computerized MC/DG could be developed. The following topics are covered:

Implementation Plan for a Full-Scale Computerized MC/DG

- User Needs/System Requirements Study
- Development of a System Design
- Characteristics of the MC/DG Data Base System
- Schedule for System Implementation
- Hardware/Software Specification
- Data Maintenance Procedures Specifications
- System Distribution Plan
- Development of a Users Guide

System Interface with a Generalized DBMS

- MC/DG Requirements for a Generalized DBMS
- Data Manipulation, Entry, Update, and Retrieval
- Data Security, Privacy, and Recovery
- Data Integrity
- Data Format Modification
- Data/Program Independence
- Data Space Management
- System Application Flexibility
- Query Capabilities
- Restrictions, Limits versus Assets

Interface with Typical State-of-the-Art Computer System

- Overall System Features
- Required Hardware Support
- Required Software Support.

This plan for a full-scale computerized MC/DG is based on the efforts made in, and the results of, the concept validation study.

2. IMPLEMENTATION PLAN FOR A FULL-SCALE COMPUTERIZED MC/DG

The following sections present an outline of how a full-scale computerized MC/DG should be developed under future contractual tasks. A tentative task phasing is illustrated in Figure 31. Hereafter, the organization selected to implement the plan is referred to as the "contractor".

a. User Needs/System Requirements Study

In designing the computerized MC/DG, contractor personnel would follow the general steps outlined in the following discussion. In many of the steps, the design of the sample MC/DG data base system and the lessons learned would serve as a guideline. The general steps for study of user needs and system requirements follow:

- Definition of the MC/DG User Group. The value of an automated data base system is primarily measured in terms of its ability to satisfy the needs of the user and its operational cost. Thus, first attention must be given to the user group and the information they need. Present knowledge indicates that the primary user group for the MC/DG will be the airframe designers. However, the influences of the interaction of the designer with other functional groups should not be neglected. Thus, review of the airframe company organization structure, objectives, and functions are also vital to definition of the system user group.
- Determination of User Needs. Once the identity of the user group is established, contractor staff would determine the information needs of the user group. The most direct approach will be through questionnaires and discussions with airframe industry designers. This should determine what types of information they have needed in the past and anticipate needing in the future. In particular, the need for data retrieval, display, analysis, and access to other (non-MC/DG) files in the computer, as well as training

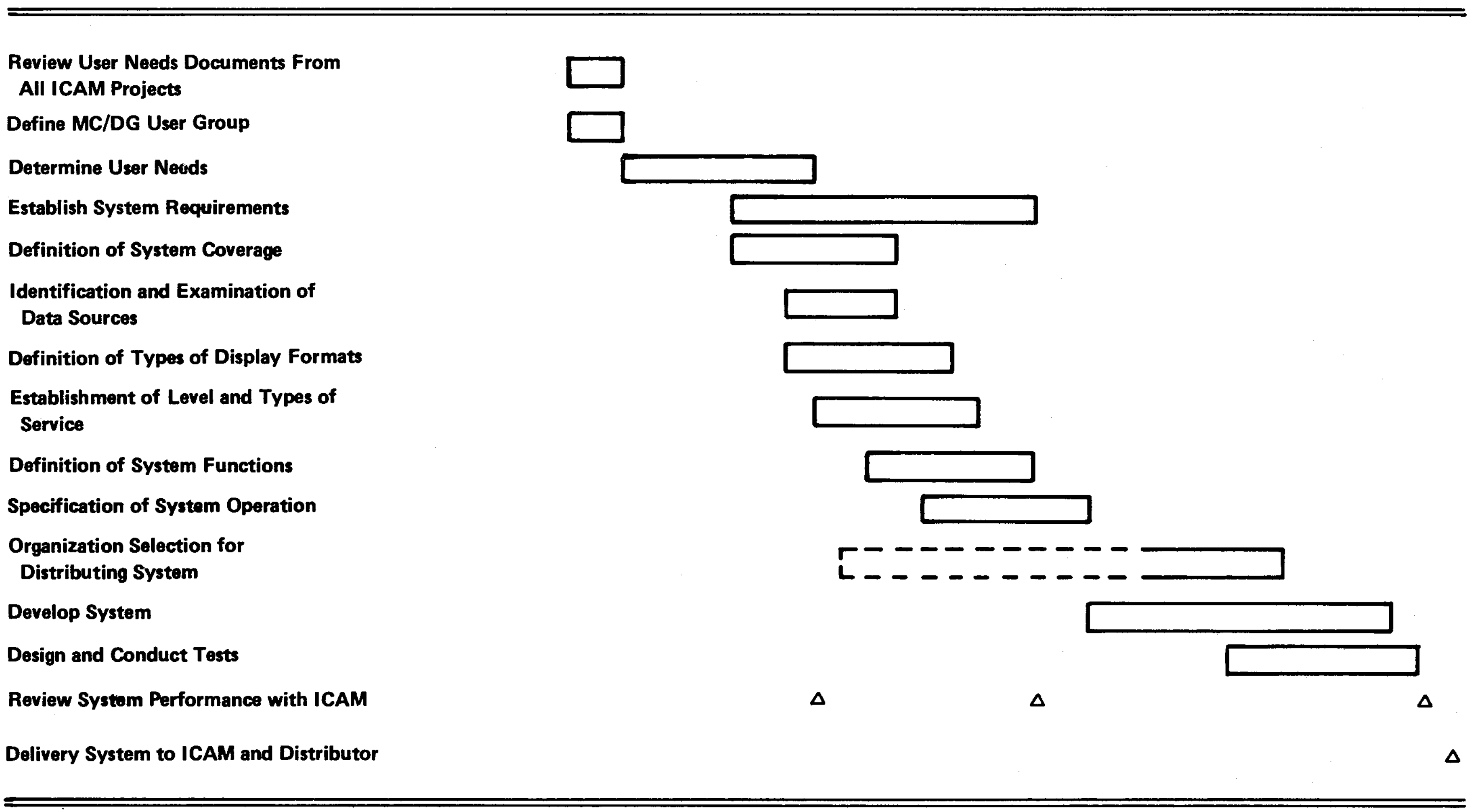


FIGURE 31. TASK PHASING OF IMPLEMENTATION PLAN

needs should be analyzed. This step represents a refinement of present knowledge of MC/DG user needs. The results should be documented using available analysis techniques such as IDEF, SADT, and SAMM.*

- Establishment of System Requirements. Contractor staff would evaluate and analyze the results of the user-needs study to establish system requirements. Particular attention should be given to the requirements for a central "Executive" model for user request processing. Also, system requirements for user request processing data retrieval, data display (tables, lists, graphs), data analysis (e.g., trade-off analysis), and interface with other (non-MC/DG) data base files (e.g., standard company parts, shapes, processing, and quality control standards) should be analyzed. These requirements should be discussed with management personnel to be certain that they are keeping with present and anticipated policies and objectives of the ICAM program. Particular note should be taken of policies and objectives that may be subject to considerable change. As an example, the interaction and cooperation with NASA's IPAD** Program should be considered. The results of the system requirements analysis should be documented using available techniques such as IDEF, SADT, and SAMM.

b. Development of a System Design

The system design for the full-scale computerized MC/DG should evolve from the design concepts developed under the present contract and from the detailed user needs/system requirements study. The general steps for system design are:

- Definition of Required Coverage (Data Elements). The data needs of the user group provides the basis for determining

---

* IDEF: ICAM Definition Method, SofTech, Inc., Waltham, Massachusetts.
SADT: Structured Analysis and Design Technique, SofTech, Inc., Waltham, Massachusetts.
SAMM: Systematic Activity Modeling Method, Boeing Computer Services, Seattle, Washington.

** IPAD: Integrated Programs for Aerospace-Vehicle Design.

the coverage of the system. Coverage is defined by a number of parameters, including (1) which aircraft parts, manufacturing processes, and materials should be included in the data base; (2) the extent of description of part shapes, usage, manufacturing process and complexities, material properties, and final condition; and (3) the detail of cost elements needed. The aim is to select limits within which the system could provide broad coverage with maximum benefit to the largest number of users. The results of this analysis should be documented, along with data base record and file documentation.

- Identification and Examination of Appropriate Sources. Once the limits of coverage have been defined, contractor staff should generate a list of specific sources from which the data could be acquired most efficiently. In most cases, this would involve examination and evaluation of data now resident at airframe companies. Continuation of the present MC/DG contract will provide much of the basic data needed for the MC/DG data base. Consideration should be given to the types of company proprietary data which could be accommodated by the MC/DG data base design when the system is installed in airframe company computers. The result of this examination should be documented and reviewed with ICAM and industry staff.
- Definition of Types of Display Formats. Using the present samples of display formats developed and the results of the user needs/system requirements study, the contractor should define the types of data display formats. The general categories of tables, graphs, and bar charts have been utilized as a part of the present contract. The results of this study should be documented by attaching samples of formats determined to be the most effective in clearly presenting the data.
- Establishment of Level and Types of Service Required. The services to be provided should be established after the determination of data coverage, data sources, and display formats. Two distinct decisions are involved: selection

of <ins>types</ins> of services and <ins>levels</ins> of services to be offered. Types of services that should be investigated are: central executive functions (request processing), data management (creation, maintenance, control of data integrity), analysis support (analytic tools for trade-offs), data retrieval, data display (tabular and graphic), user training, and user-program integration and management. As an example of the <ins>level</ins> of service required, several levels of data management should be examined: primary MC/DG data management, user auxiliary data management (perhaps including subsets of other data bases), and interface with (access to) other data bases for such data as standard parts, stock materials, shop floor schedules, etc. This question of level of service highlights a bigger question of how much the computerized MC/DG should be integrated with other company functions such as planning, manufacturing, purchasing, program schedule, cost control, etc. The current philosophy of the ICAM Program Office is that these company functions should be performed by an integrated computerized system. The results of this analysis should be documented and reviewed by ICAM and airframe industry staff.

- <ins>Definition of System Functions</ins>. Once the types and levels of services have been specified, the contractor should determine the basic system functions necessary to provide these services. As a part of this determination, it will be necessary to determine which functions should be performed by existing computer systems/subsystems (i.e., which functions should be performed by vendor-supplied system support software and/or by the GDBMS and which functions should reside in the MC/DG system). This study should build upon preliminary system function definitions established as a part of the current contract. The results of this determination should be documented in detail using diagramatic illustrations of the hierarchy of system functions.

- Specification of System Operation. This step in the system design goes beyond the naming of functions to be performed; instead, it should describe how the named function could be performed. For example, if the function is to access data, the access method should be specified. In all cases, alternative methods should be weighed in terms of their compatibility with other features of the system. The system operation specifications should be subjected to a detailed system design review by the contractor and then documented at the levels appropriate for programming and for system maintenance.
- Provision for Monitoring and Accounting. System operation monitoring and accounting are actually system operations to be specified above; visibility as a separate task is given here because of the importance of these system functions in providing feedback data for evolutionary improvements of efficiency when the system becomes operational.

c. Characteristics of the MC/DG Data Base System

- Large amount of data that is fairly static, i.e., does not require frequent update (replacement of old data with new data); hence, it should be retrieval oriented.
- More emphasis on textual term retrieval than on numeric term retrieval, i.e., the capability of content search, full text inversion, stemming, display of related terms, and support of variable length text data fields.
- Need a self-contained system so that the general user need not be burdened with writing his own application programs, but rather, simply invoke system commands to retrieve, analyze, and display his data.
- Both predetermined transaction processing and ad hoc query may be required. However, the majority will most likely be predetermined. In most DBMS's, unanticipated quarties, or ad hoc query requirements, usually have to pay the penalty of a time-consuming sequential scan if the

needed data elements are not keyed or inverted. BASIS handles that efficiently since it has inverted indexing capability.

- Transaction-oriented functions such as frequent request for Cost-Driver Effects (CDE) and Cost-Estimating Data (CED) displays should be predefined so that users need not write his own program to do that. The PROFILE module of BASIS provides those capabilities.
- Nonhierarchical type of data that requires nonnetwork structures. Data record for each part is indepenent of another.

d. Schedule for System Implementation

Upon completion of the User Needs/System Requirements Study and the Detail Design Tasks, the system implementation should be scheduled. This effort should consist of the following:

- System Development. The designed system must be developed according to the system design prepared. This involves:
  - Coding, debugging, testing, and documenting subroutines and programs, and interfacing modules
  - Preparing detailed procedures and operating ground rules
  - Writing (coding or specifying) detailed formats
  - Determining operational support requirements (staff and other resource needs)
  - Writing training guides and programming on-line macro tutorial procedures
  - Writing system documentation suitable for system maintenance by support staff.
- Design and Conduct Tests. The developed system should be thoroughly tested and identified deficiencies corrected before the system is made operational on a customer's computer. Suitable test procedures must be designed and implemented. These tests would evaluate, in detail, individual module performance and module interface performance.

- Review System Performance with the ICAM Program Office and Market System to Customers (Airframe Companies). Concurrent with designing and conducting testing, a final system review should be scheduled with the ICAM Program Office. At about the same time, initial marketing of the system to the airframe companies should begin. Draft promotional and system documentation should be presented at the ICAM review meeting.
- System Delivery. Upon system acceptance by the ICAM Program Office, the system should be delivered to the organization chosen for system distribution (installation at customer sites). See the following section on a Full-Scale Distribution Plan.

e. Hardware and Software Specifications

(1) Hardware Specifications--Minimal Requirements

The full-scale MC/DG system will need to be implemented in a medium-to-large current state-of-the-art computer system. The final selection depends heavily on the amount and complexity of the data to be stored in the MC/DG data base, as well as on the number of users and their mode of accessing this data base. The following is a list of minimal hardware configuration recommendations for a full-scale MC/DG system as envisioned at the present time:

- Central Processing Unit (CPU). This controlling center of the computer system should provide facilities for:
  - Addressing main storage
  - Fetching and storing data
  - Arithmetic and logical processing of data
  - Executing instructions in a desired sequence
  - Initiating communication between main storage and input/output (I/O) devices.
- Main Core Storage (MC). The main storage provides the system with directly addressable, fast-access storage of data. Both data and programs must be loaded into main core (from input devices) before they can be processed. Minimum main

storage capacity of approximately 3000K bytes is recommended. The actual core capacity requirement depends completely on the host computer system and the DBMS that the MC/DG is interfaced with.

- High-Speed Buffer Storage. Buffer storage can sharply reduce the time required for fetching the currently used section of the main storage. Buffer operation is handled entirely by hardware and is transparent to the user, who does not need to adhere to any particular structure in order to achieve close-to-optimum use of the buffer.
- Input/Output Equipment. An input/output operation transfers data between main core and an I/O "device". An I/O operation is initiated by a program instruction that generates a command to an I/O "channel". A "control unit" receives the command via the I/O "interface", decodes it, and starts the I/O device.
- I/O Devices. Fall into a number of categories. They are required for:
  - Auxiliary storage, e.g., disk, tapes
  - Machine and manual (keyed) input, both local and remote, e.g., keypunch, key to disk
  - On-line terminals, e.g., Silent 700
  - Reading/printing/displaying of external document and graphic displays, e.g., card reader, line printer, CRT.
- I/O Channels. Are the direct controllers of I/O devices and control units. They provide the computer system with the ability to read, write, and compute simultaneously, by relieving the CPU of the task of communicating directly with the I/O devices. Channels may be stand-alone units, complete with the necessary logical and storage capabilities, or they may be time-share CPU facilities and be physically integrated with the CPU.
- Control Units. Provide the logic circuitry and the storage areas (buffers) needed to operate the attached I/O devices. A control unit may be single path which controls only one device, shared-path, or multipath, which permits several I/O

devices to transfer data concurrently. Those with multi-path are recommended for fast response time in accessing the MC/DG data base.

(2) Teleprocessing Requirements

The teleprocessing system must meet the following requirements. It should be capable of servicing many users at different locations, some on common communication lines, and some on separate lines. The transmission control equipment and programming must be able to handle the multiple inputs arriving in unscheduled fashion into the computer system. The circuits (channels or lines) that transmit information between the terminals should preferably be duplex circuits, i.e., they carry data in two directions at the same time as opposed to the simplex circuits (they carry data in only one direction) or the half-duplex circuits (they can carry data in two directions only one at a time). The mode of transmission for these circuits should be parallel, since parallel transmission allows all bits of a chracter to be transmitted simultaneously.

(3) Software Minimum Requirements

The operating system--the collection of software (programs) that organizes the processes and peripheral devices into a high-performance application execution system. The operation system's basic components should include:

- Processes that control initial resource allocation, communicate with the system operator, and log errors
- The command interpreters
- User-programmed process control services
- Exception dispatcher
- Memory management routines for program image activation and paging
- Scheduling routines and swapper
- Interrupt and input/output processing routines
- Compatibility mode execution routines.

The resources of the computer system are the CPU, core memory, and the peripherals. The system handles many jobs simultaneously, and each job can have different resource requirements. The operating system

enables jobs to share the resources according to their individual needs, and also protects each job and its data from other jobs on the system. These tasks are performed via the scheduling, memory management, device allocation, and I/O processing modules within the operating system.

It is extremely important that the host computer system be operative under a full operating system with the aforementioned functions in order that the MC/DG system can be successfully implemented. It is also essential that the operating system supports time-sharing teleprocessing so that users will have the option of accessing the MC/DG data base from remote locations via terminals.

- The languages provided by the host computer system should include the major scientific application oriented high-level programming language, i.e., FORTRAN. Other high-level languages like PL1, COBOL, and APL may be useful if users wish to write their own application programs to interface with the MC/DG system. Many existing general DBMS's either accept FORTRAN as a host language or utilize it themselves.
- The utility library is a collection of special purpose programs that are callable by user programs. They may be used to perform mathematical/statistical functions, for sorting/merging large data files and, in general, for data manipulation. It is essential for the MC/DG system to provide or support such packages (e.g., IBM Scientific Subroutine Package, and Statistical Package for the Social Sciences).
- Data Management Methods. Should provide these services:
  - I/O device control
  - File access method(s)
  - Record management services
  - Command interpreter and utility program.
- The I/O device control does the basic I/O device handling for all of the other data management services.
- The file access method provides flexible, efficient data management for disk volumes (e.g., direct, random access, index, and/or sequential access methods) and magnetic tape volumes (sequential access method).

- The record management services are a set of system procedures that provide efficient and flexible facilities for data manipulation and storage. It is essential that the host computer system provides such capabilities to ensure the integrity of the MC/DG data base.

f. Data Maintenance Procedures--Specifications

(1) File Structure Definition Method

The full-scale MC/DG system will encompass a vast amount of data. Thus, it is important that the contractor exert great care in the design of the data files. Results from the user needs/system requirements study and lessons learned from the concept validation study of a computerized MC/DG should be taken into careful consideration. It is desirable that the MC/DG data base file structure be defined and created by a self-contained Data Definition Language, independent of the host computer system hardware/software configuration. At the present, there are several general DBMS's available on the market that are operative under different computer systems. It is the contractor's responsibility to select the one that is most suitable for the MC/DG system and can be operative on most of the computer systems currently utilized by airframe companies.

(2) Data File Creation

Once the file structure is well defined, detailed data file creation procedures should be developed. Packaged software modules supplied by off-the-shelf DBMS vendors may be used for the actual creation of the data base files. Catalogued procedures which group the various functions and system utilities, as well as the necessary job control language into a unified job step, should be created for consistent and efficient data base maintenance (update, add, delete data record, and/or data elements). These procedures may be stored on the computer for fast, on-line execution. They should be carefully documented to ensure their validity and proper function.

(3) Data File Recovery Procedures

The procedures must be carefully planned, implemented, and documented. Care must be taken to guard against data loss or contamination due to system crash or change. A well-documented and regularly executed data base back-up procedure must be implemented. In case of accidental loss of data files, prompt and accurate recovery procedures must be provided to restore the data base to its former intact state.

g. System Distribution Plan

Once the full-scale computerized MC/DG system has been developed and tested, a plan should be ready for the distribution mode. The key issues to be decided upon are:

- Who (what organization) should distribute the developed system?
- What incentives exist for the selected organization to accept distribution responsibility?
- What contractual arrangements should be made between the distributing organization and the customer organization?

A discussion of these issues and our recommendations follow in the remainder of this section.

(1) Selection of an Organization

Candidate classes of organizations to be considered as distributors of the developed full-scale MC/DG system are:

- The U.S. Government (the USAF/ICAM Program Office is paying for development)
- The contractor selected to develop the system (this contractor is in the best position to maintain the system--correct problems and implement enhancements)
- Other contractors with a demonstrated ability to promote a product, train users and system support staff, and to maintain software and documentation
- Professional societies with an interest in promoting the general area of computer-aided manufacturing

- Consortium of users (customers) within the airframe industry
- A newly established, perhaps government subsidized, organization whose mission would be effecting technology transfer from government to industry.

It is recommended that a contractor be selected by competitive source selection. The criteria for selection should be the contractor's ability to maintain, install, and market the system (software and documentation) and to provide system training services. The contractor should have experience in system design and implementation on a variety of host computer and operating systems now being utilized in the airframe industry.

(2) Incentives

The organization selected should have financial, technical, and public relation incentives for wanting to perform system distribution functions. The contract for distribution should give adequate profit incentives. The technical and public relation incentives can be provided by the opportunity to provide challenging state-of-the-art support services (system maintenance) and training to the U.S. Air Force and a number of large airframe companies.

(3) Contractual Services

The organization selected for system distribution should provide full technical and training support, under direct contract to the customer, for installation (and optionally, continuing maintenance) of the computerized MC/DG system. The basic installation services contract should allow reasonable fee or profit to the distributor. Also, the opportunity should be open for the distributor to provide additional services to the customer as might be mutually agreeable. Such services could include the following:

- Continuing system maintenance
- Recurring training of new staff
- Development of user application modules
- Development of interfaces to unique company systems and modules.

The distributing contractor should make available, to all customers, a basic installation package at a price schedule to be agreed upon in the distributor's contract with the ICAM Program Office.

h. Development of a Users Guide

(1) Systems Users Guide

In order that the full-scale MC/DG system be effectively and satisfactorily utilized, the development of a full-service users guide is mandatory.

Such a guide should be:

- Comprehensively Written. The procedures (the commands with their associated parameters) should be clearly specified and enumerated such that their connotations are nonambiguous to the users, e.g.,
  - For a batch (off-line) computer run, specific and comprehensive job-control language and job-deck setup should be well documented
  - For on-line access mode, the logon (login) and logoff (logout) procedures, including dial-up instructions, user ID, and password requirements to access the system must be included in the user guide.

  In addition to hard copy instructional material such as the users guide, the user should have access to on-line user aids and instructions (see previous sections on system requirements, particularly the System Training Aids and User Instruction Module).
- Self-Explanatory. Providing optional tutorials upon user request. Often, the terminology and context described in a users guide may not be immediately obvious to a noncomputer-oriented user. It may become impossible for him to proceed further from a certain point of MC/DG system application. He is in need of further explanation to effectively continue utilizing the computerized system. It is at this moment that an optional, more lengthy tutorial would be extremely helpful, e.g.,
  - After logon and acquisition of the proper data base, the system may type out "ENTER YOUR REQUEST"
  - The first time user is faced with the puzzle "HOW?", some form of explanation will be given.

3. SYSTEM INTERFACE WITH A GENERALIZED DBMS

a. Definition of DBMS

A data base is generally defined as a collection of information specially organized for analysis or is used as the basis for a decision. This collection may be stored on drums, disks, or other secondary storage media. The data base is integrated; that is, it contains nonredundant data for not one, but many, users for varying purposes.

A data base system generally has a set of ordinary batch application programs which access the data base--retrieving, updating, adding, or deleting the data. Additionally, there may be a group of on-line users who interact with the data base from remote terminals, performing the same type of operations as the batch application users.

Pictorially, a data base system may be illustrated by Figure 32.

b. MC/DG Requirements for a Generalized DBMS

It is difficult to find an off-the-shelf DBMS that meets all needs and is exactly tailored for an intended application. Each system is built with certain objectives, hence, it has specific capabilities for the intended application. Prior to selecting a system, managers must make a definitive analysis of their requirements specifically noting which features are mandatory and which are desirable. The following features deserve careful consideration:

- Numeric data versus textual data oriented
- Retrieval oriented versus update oriented
- Self-contained language versus host language
- Predetermined transaction processing versus ad hoc query
- Procedural programming language versus predefined process via user language macro
- Network versus non-network structures.

c. User Requirements of DBMS

There has not been a universally accepted set of characteristics which define a DBMS. The technique has undergone constant evaluation during the last decade. However, there are numerous criteria typically

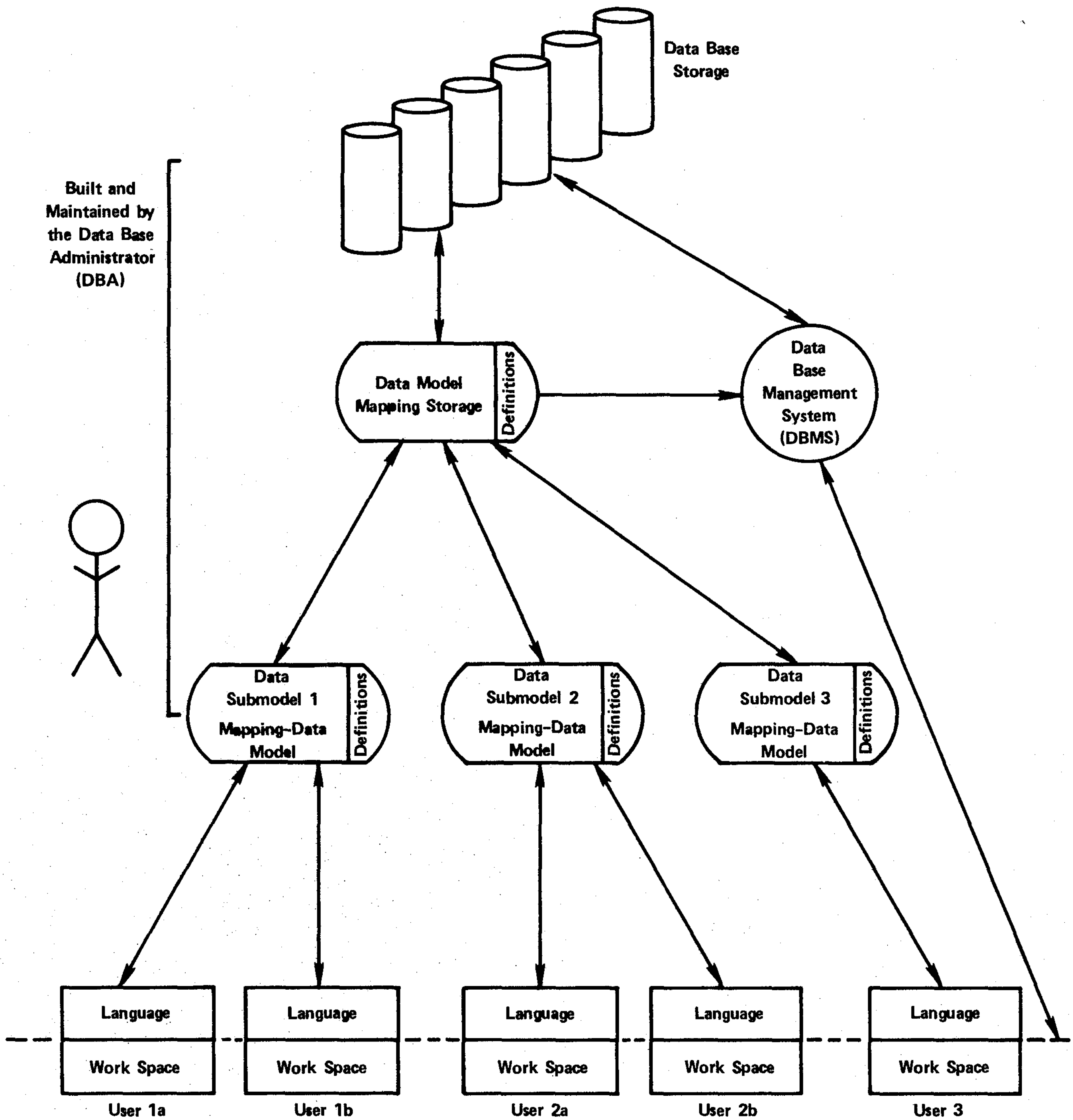


FIGURE 32. AN ARCHITECTURE FOR A DATA BASE SYSTEM

used to specify what benefits are required from a data base system. They can be generalized to the following:

- Response-time requirements--batch and on-line
- Storage requirements
- Maintenance requirements
- Inquiries, updates, reporting capabilities
- Data nonredundancy
- Data reliability and flexibility
- Data security
- Language and application independence
- Transportability
- Economic feasibility.

The first three requirements can be quantified and, hence, are measurable; whereas the others are increasingly qualitative and, therefore, are subjective. Consequently, satisfying user requirements of a data base system relies heavily on the hardware and software of the host computer system.

Desirable, but not absolutely necessary, are the following additional criteria:

- The ability to perform logical sequential processing based on the values of a key field
- The ability to build intrafile record relationships based on information derived from those files
- The ability to establish some form of ordering on such relationships
- The ability to create, as well as eliminate, such relationships while the file is being accessed by other users
- The ability to perform Boolean and numeric searches on such relationships
- The ability to handle missing values for numeric fields, without special encoding
- The ability to distinguish blanks from non-blank alpha fields
- Data retrieval can be accomplished via such mechanisms as:

- Disk order
- Logical sequence
- Intrafile relationships
- Field value.

(1) Selection of DBMS

Due to the vast selection of generalized DBMS's available on the computer market today, prospective users are faced with the dilemma of which one to choose. There is no simple answer to the question "Which DBMS is the best choice?" because of the advantages and disadvantages needed to be considered in light of an installation's ultimate need. In order to assess the effectiveness of a DBMS, the following management type of questions need to be considered:

- What are the basic requirements and services that a chosen DBMS can offer?
- Is the system easy or difficult to install?
- How much time will be required for the user to learn to use it?
- What skill level is needed to operate, maintain, and use the system?
- On what computer configuration(s) can the DBMS be installed?
- How many application programs are necessary?
- How good are the system documentation, users guide, and training provided?
- What type of support does the vendor provide in installation, full implementation, and maintenance of the system?
- How cost effective is the system?
- How flexible is the system in terms of growth and expansion.

The information contained in Table 2 was quoted from the article "Data Base Systems: Design, Implementation, and Management" by R. G. Ross in the June 5, 1978, issue of Computerworld.

TABLE 2. DBMS BY VENDORS AND NUMBER OF USERS

System	Vendor	Estimated No. Of Users
ADABAS	Software AG	200
IMS	IBM Corporation	600-900
IDMS	Cullinance Corporation	200
System 2000	MRI Systems Corporation	200
TOTAL	Cincom Systems, Inc.	1,100

A DBMS typically "manages" the data base. That is, it is designed primarily to "organize" the data sets or files that constitute a data base. The data manipulation and retrieval operations made available in the DBMS packages may be invoked or called in many different ways. The methods employed range from specification statements included directly in the syntactic structure of the host language, to macro calls that reference vendor-supplied subroutines. The relative advantages of these approaches are quite user-dependent since the natural extensions of these language specifications are more or less limited by their structure. Most DBMS packages available today require a high level of user skill. In general, the user who prepares a query in the host language must have a skill level that at least provides him with a programming capability in the host language. The vendor generally does not supply users with application programs. Complex programs may be required to perform hueristic searches, i.e., searching for data elements/records on the basis of data previously retrieved from the data base. From an external point of view, this may imply a capability to "browse" through the data base. For the application program, it may mean a capability to iterate and recursively employ the results of previous data base references within the user program.

(2) Synopsis of Five Major DBMS

Note that each DBMS was originally designed for a certain set of users in a given system hardware/software environment. It is understandable that their functions would be most effective under those circumstances. The following section presents a systematic synopsis of five major DBMS's:

(1) ADABAS: Adaptable Data Base System
(2) IDMS: - Integrated Data Base Management System
(3) IMS: Information Management System
(4) S2000: System 2000
(5) TOTAL: Total System.

(3) Data Structure

<u>ADABAS</u>. ADABAS is a full-scale data base management system. It excels in medium-size to large application environments. It is

comparable in general data-handling capabilities to any of the other major data base systems. ADABAS has a single level file structure with Repeating Groups (RG). Variable length text fields, as well as multiple fields, are allowed. Multilevel hierarchical network relationships can be achieved by coupling two files together. Under ADABAS data structuring, all logical data element relationships are given physical expression in a segregated portion of the data base. These inversion techniques result in a considerable reduction of overhead associated with file organization and data base searching. A record in an ADABAS data base is a simple physical hierarchy in which an effective limit of one or two levels is possible. The capabilities of indexing these records and the mapping of indexes for elements of the same type appearing in different records are very useful.

IDMS. Integrated Data Base Management System (DMS) was originally designed as a single batch system. A multi-user capability is available through a facility called Generalized Communications Interface (GCI). The system includes two languages for definition and manipulation of the data base: a data description language (DDL) and a data manipulation language (DML). Physical storage of the data is in fixed length Basic Direct Access Method (BDAM) blocks called pages, whose lengths can be defined by the user at system generation time. Within the pages, data is logically available to the user as records. At schema definition time, the user will define all types of records available in his system, including the specific characteristics of each field within the record. Each page contains an array of pointers and descriptors which reference the record on that page. Physical access to the data base is accomplished by bringing the page into the IDMS buffer area. IDMS then locates a specific record through the use of the pointer array of each page.

IMS. IMS processes only hierarchically structured data bases. The data for the physical entry are distributed through a series of levels, each of which is comprised of one or more segments. These data segments are logically grouped together via a set of pointers. The sequence of placement within segment groupings is always top to bottom, down a single

hierarchical path, with the various paths stored left to right (represented schematically by tree structures). A number of choices in internal pointer types and storage patterns are available. Entry point access to data base roots is sequential, randomized, or indexed.

System 2000. System 2000 is an inverted list type of data base management system that processes hierarchically structured data bases, with full inversion on selected data elements. The system's data base description method defines the logical files of the data base. The basic data item is called an element which can be designated a key value to be used as a search criterion. A collection of elements form a Repeating Group (RG). It occupies a specific level in the hierarchical tree and may be related to other RG as an "ancestor" or "descendant". Each occurrence of an RG and its elements is called a "data set". System 2000 constructs it physical files in a serial form using data in the data base description to construct various types of reference tables.

TOTAL. The TOTAL data base management system can be described as a partially inverted system organized into a network of file structures. Its inverted list is distributed as linkages, chains, or record pointers within the records themselves. The records are fixed length and are categorized by the type of data file to which they belong, single-entry files or variable-entry files. Single-entry files contain the master key data for a cohesive information set that is distributed through the variable-entry files. Their entries are positioned randomly on the basis of a designated key value. Variable-entry files are organized serially. They are logically linked to other similar entries in the variable-entry file, as well as to the base single-entry record to which it belongs.

(4) Data Manipulation, Entry, Update, and Retrieval

ADABAS. ADABAS is a self-contained system in that it provides all essential capabilities for data creation, data update, data retrieval, and report formatting. Input data validation is performed by matching data types, checking preprogrammed range values, and verifying data conversion. The update language is similar to the query language, Adascript.

Update values may be specified as the results of a computation, and may be performed at the data element level, on the items within repeating groups, or at the record level. These system-triggered updates include adjusting all pointers involved.

IDMS. All data manipulation functions on IDMS data bases take place at the program interface level, whether through the use of the DML preprocessor code or through specifically coded CALL statements. There is no stand-alone or natural language by which the data base may be accessed. Therefore, the user of IDMS must at least be familiar with COBOL. The user needs to set aside portions of his program's work space for retrieval, building, and storage of the various record types with which he intends to work. The user has a series of DML commands at his disposal. The INSERT and STORE commands are used to add new records and/or establish additional set relationships. The MODIFY command allows the user to change the contents of one or more elements within a record. The REMOVE and DELETE commands allow for the user to remove records from specific set relationships and/or from the data base entirely. It is essential that the user understands thoroughly the relationships between the data sets within the data base in order to update any of its records.

IMS. The data manipulation and retrieval operations for IMS are performed through a combination of functions and facilities known collectively as Data Language 1 (DL/1). The potential for data manipulation is generally predetermined at data base generation time. All references to IMS data items are made through the facilities of DL/1 which is activated via the available call function of the host language (COBOL, PL1, or BAL). Each DL/1 call contains the DL/1 function to be executed, plus a variable number of parameters and a parameter count field. DL/1 uses this parameter set to determine the set of control data which must be referenced in order to gain access to the data of interest. Data items can be entered at the head, tail, or middle of a logical linkage chain within the hierarchical sequence.

System 2000. System 2000 has two methods of processing data:

- The natural language processing can be performed in both batch or Teleprocessing (TP) mode, using either queried or immediate access techniques.
- The procedural language processing can be run in batch mode only. It permits the user to operate in COBOL or FORTRAN host language environment and provides means for manipulating and querying System 2000 data bases within the context of these languages.

The system provides full entry, update, and delete facilities for each of its configurations. In the natural language processing, an entire search and update operation to be performed on the data base will be specified within one syntactic unit and no reference will be made to positioning which has resulted from a prior command. Procedural language operates on one data set at a time and relies on the position within the data base, established by prior operations to determine the effective context of any further operation. Thus, the procedural language user has more control over his position in the data base.

TOTAL. All data entry, update, and deletion operations are accomplished through the facilities of a Data Management Language (DML) that interfaces with a host language such as a COBOL, PL1, or assembler language. The DML is structured so that each command consists of a call followed by a set of parameters. The commands are organized so that data may be accessed serially or randomly, using a key to randomize the physical location of the record of interest. TOTAL accesses individual data fields as opposed to records, the fields to be accessed may be retrieved or stored in an arbitrary order. Records added to a single entry data set are mapped into a data space on the basis of a key-key value pair that is defined by the data base definition language. Records added to a variable entry data set may be appended either to the beginning or the end of a sequence of occurrences of common entries.

(5) Data Security, Privacy, and Recovery

ADABAS. The ADABAS user is provided data security at the data base level, the file level, and the data element level. However, data

security is not guaranteed at the TP terminal. Up to 15 levels of data security may be assigned. User password is translated to an access-level number and an update-level numbers. If the data element accessed has a higher level number than the user's access-level number, then this data element is locked to that user. The system has tapes for backup and recovery. Other recovery features are:

- An autorestart capability for recovery from hard read errors
- A restore capability which restores before-images, following abnormal program termination.

IDMS. Under IDMS, data privacy facilities are implemented by means of the subschema facility. No user may request data management services from the system without a designated subschema. This restricts his universe of concern to particular areas, data items, record types, and sets. In the area of data security, IDMS permits subschemas to be defined as read only. The pointer information which is physically prefixed onto the front of each record occurrence is stripped off by IDMS before the requisite data items are moved to the user's work area. These are not made available to the user, except indirectly. Overlaying or alteration of these values is not permitted.

Recovery of the IDMS data base is possible in either a forward or backward direction. The system includes a security dump utility program which allows the user to backup all or portions of the data base periodically. A security restore program permits reestablishment of all or part of the data base to a desired date and time.

IMS. Data privacy and access authority (retrieval, update, replace, or delete) options are declared during data base generation. The security of the data base is protected in part by restricting the user from having access to the structured data of the files. Recovery protection is provided in the form of standard logging operations which record both the contents of the segment before update and the operations performed. In case of system failure, a set of utility programs is available for reprocessing an archival copy of the data files against the log files. IMS/VS provides an optional batch checkpoint/restart facility to

provide long batch programs the ability to coordinate recovery with their data base processing.

System 2000. The system uses a password concept coupled with an access authority. A master password is created for a data base at definition time. The system may choose to assign one or more passwords and specify a range of access authorities for each password. A password may be assigned READ ONLY, UPDATE, or QUERY authority for each element in the data base. Passwords may be added, deleted, or modified at any time during the data base's life. However, only the master password holder can request that the data base be restored in case it has been "damaged" during an update. There is a security feature in the IBS system called "security by entry". Using the master password, one element in the entry RG may be selected as an entry key. Thereafter, all secondary passwords may, at any one time, have access to only one logical entry. This kind of security can be a very useful feature in a data center type of application wherein the data center provides one logical entry for each user's data.

TOTAL. No specific provisions are defined within TOTAL for privacy of data. They are embedded in the Data Management Language (DML) itself. For each retrieval call, only those elements or fields specified within an element list parameter are returned to the user. Additional privacy facilities are solely the responsibility of the user and must be accomplished administratively or through embedded provisions of the host system.

Security is accomplished in TOTAL by not granting the user access to the structured data that links the entries. The user is responsible for the integrity of the attribute values used in creating the structured data for the various linkage paths.

TOTAL provides fairly adequate recovery procedures for recording the data necessary to restart the system from a known point and to ensure the integrity of the data base. This is accomplished by means of dynamic data base logging of before-and-after update images.

(6) Data Integrity

ADABAS. An ADABAS data base is physically and logically separated into two major areas, namely, the DATA STORAGE and the ASSOCIATOR. All control information for the data base is kept in the ASSOCIATOR. The logical data records contain all existing values for all the fields and are maintained in the DATA STORAGE area. Users may specify null value suppression to differentiate between fields for which a null value implies an empty field and fields in which a null value implies a zero if it is numeric or blank if it is textual.

Since the user may, at any time, override the standard length or standard format of a field by explicitly requesting a different length or format within the ADABAS command that reference the field, the integrity of the data becomes the sole responsibility of the user.

IDMS. A user must declare his intentions for the use of an area when he opens an IDMS data base. An area may be opened for any of these modes:

- Simple update
- Simple retrieval
- Protected update
- Protected retrieval
- Exclusive update
- Exclusive retrieval.

Each of these modes implies a degree of protection for the user in the interactive environment. While operating in update mode, the subject areas are protected from other run units. The exclusive options bar any other user of these areas from access while the run unit is in operation, regardless of the other user's update intent.

The system assumes sole responsibility in the maintenance of all pointers and other structured data.

IMS. Although IMS defines a parameter of the FIELD specification whereby data type can be declared within its data definition language, it does not provide any object time checking for incorrect data format. If data of one type are inadvertently stored into a field defined for another type, the error may never be discovered.

System 2000. During processing, the system performs all necessary conversions for any of the six system data types: NAME, TEXT, INTEGER, DECIMAL, MONEY, and DATE. The first five may be described by a picture designation defining its length attributes. Both DECIMAL and MONEY may define a decimal point within the picture. Errors in attempting to perform arithmetic operations on nonnumeric fields are checked. Overflow on date or numeric items are not permitted. Input data are checked for validity of form. Character type data may exceed its defined length. Its excess is stored in an overflow record.

TOTAL. Data integrity at the data item format level is the user's responsibility. The user is to ensure that mixed data are not placed in a defined field, e.g., decimal data are not inserted into a field defined as binary. TOTAL performs no checks on data format. At the data set level, TOTAL checks validity of logical data structure and linkage chains except in the case of serial-write operations, in which case no structural maintenance is performed.

(7) Data Format Modification

ADABAS. The data format type supported by ADABAS are: alphanumeric, unpacked decimal, packed decimal, binary, and fixed point. The user may modify the data field value and format via the commands UPDATE and ADD. The new value of a field may be shorter, of the same length, or longer than the current field value. Any required data format conversions are performed by ADABAS.

IDMS. IDMS does not implement any degree of data format modification. The user receives the data in the same format as it has been stored in the data base. It is the user's responsibility to ensure that the actual data content within each of the described fields of a record has the proper data format when the record is stored or modified.

IMS. Since no checking is done on the content of a data field, relative to its declared format, no provisions exist with IMS for format modification within a data field. If a new segment is to be defined in an

existing data base or an existing segment to be restructured, the primary effect is only on those applications using the new or redefined segments. If new segments are to be processed by an application, new processing algorithms must be added; the program communication block and the program specification block must be modified as well.

System 2000. Some data format modification is allowed. New repeating groups may be added to the bottom of a hierarchical structure without affecting either the program or the file organization. However, if either an element within an RG which has existing data sets, or if the hierarchical structure or replacement of the repeating group is modified, the file must be reorganized. Adding elements to existing RG's will not require a change in the user's programs.

TOTAL. TOTAL does not provide dynamic format modification capabilities. If the data of one type extracted from File A is to be placed in a field defined for another type of data in File B, the necessary conversion and formatting must be done by the user. At the level of the data base description, record formats may be modified through the data base definition language without affecting the programs that use the modified formats. It will be necessary to modify the user programs only if elements or fields currently referenced by a user are modified.

(8) Data/Program Independent

ADABAS. ADABAS provides a total automatic restructuring capability to modify existing data structure via a command language. Users may add or delete single data elements by using the appropriate command. The system will add or delete the element from the data definition table. Values for each data are added or deleted using normal update commands.

IDMS. When data fields are added to an existing record type (or removed, expanded, etc.), the schema definition is recreated and the appropriate subschemas regenerated. New set relationships may be added between existing record types without affecting existing application

programs. New applications may be added completely separate from, or related to, existing data structures without recompilation. However, if a user or applications run unit uses a new relation, data element, or record type that is being created, the program involved must be recompiled. Programs which perform DELETE or STORE operations must also be recompiled when records, to be stored or deleted by those programs, are redefined to be coupled with new records of the subschema.

IMS. An IMS data base is really comprised of two parts:

- The physical data base is represented by the data base definition and the access method selected for the data base
- The logical data base consists of logical structure data residing in the physical data base.

The degree of data independence is a function of the data base definition of the various physical data bases. It depends on the skill and imagination of the data base designer and on the level of general data analysis performed prior to structuring the data bases themselves.

System 2000. Data/program independence is achieved in System 2000 through its various data storage techniques. Physical storage is totally transparent to the application programs. Through the system's data definition processes, a series of tables is constructed which is used to reference arbitrarily stored data sets. With the procedural language processing, users are required to establish a universe of data which delimits the elements that can be retrieved in a given operation. If the original hierarchical structure has been modified so that single RG's or entire tree structures have been added to the data base at a logical position other than the end of the data base, the affected application programs may require adjustments (i.e., parent/descendant relationships have been modified).

TOTAL. TOTAL does not retrieve data records but rather, only those fields or data elements that are specified as parameters of the calling sequence in the application program. This implies that application programs are independent of field order within a record. From the structural point of view, TOTAL provides a degree of data/program

independence because all references to records in variable entry files are by key-key value pair with respect to a specific single entry file. Provided that in any record modification, the key for the associated single entry file is preserved, the corresponding application programs are not affected.

(9) Data Space Management

ADABAS. Only fixed length physical records and blocks are handled by ADABAS. The data storage portion of the data base contains the data records for all user applications. The system stores only the meaningful field values in the data records. Empty fields, leading zeros in numeric fields and trailing blanks in alphanumeric fields are not stored. The compressed data record in data storage contains no field names, no address pointers, no overflow pointers, only meaningful data. A single physical block contains many variable length logical records (in compressed form). A certain percentage of each physical block is reserved for efficient handling of possible expansions of logical records within a block. This compression technique can result in substantial savings in overall physical storage requirement.

IDMS. There is no specially designated "overflow area" in the IDMS data space. All pages within a particular area are potential storage locations for a record occurrence assigned to that area. That is, when the data base gets very full, the system will accept any page within a record's assigned area for storage of an occurrence. Space management within the IDMS environment focuses on so-called calc records, their placement, maintenance, and retrieval. Calc records are the type of records placed in the data base on the basis of a randomization on a specified key field within those records. When the system attempts to place a calc record in the data base for the first time, the randomization is done only to a specified page in the data space. Provided the designated page has sufficient space, an index entry is created for the occurrence in the page, the record is stored, and the new data base key is returned to the user.

IMS. IMS data space management requirements depend largely on the access method for the data base. For Hierarchical Index Sequential Access Method (HISAM), physical data base records are stored sequentially according to a user-defined key, relating to the root segment. New records are added in the proper physical position, according to the values of the sequencing key. This requires unloading and loading the physical data base. For Hierarchical Direct Access Method (HIDAM), two data spaces are required. One is used for tabular indexing, the other for the storage of the physical data base record. Records stored in the entry sequenced data set are placed in physical sequence according to a user-specified key. New records may be added at the end of the data space.

System 2000. System 2000 has one basic file organization from which the system constructs a variety of internal tables and formats that are transparent to the application program. The space required to store the data base is partially a function of the size of the generated inverted lists. Each System 2000 data base has six files: data base definition table, unique value table, values entries table, overflow file, hierarchical location table, and the data file. All these files are organized by the Basic Direct Access Method (BDAM) of IBM. The size of these six files are subject to user definition and must be calculated by the user, either from a set of manual techniques or from a program (both provided by the vendor). The actual size allocated to the data base may be modified at any time merely by performing a "SAVE" function of the data base, deallocating, and reallocating the files, then performing a "LOAD" function.

The data file consists of variable length records representing occurrences of all RG types. The record distribution method for this file is arbitrary. The distribution algorithm is designed, together with the space management technique, to keep all the records of a complete logical entry physically together. This enhances efficiency at data base loading time.

TOTAL. All records in a TOTAL data base are of fixed length. Hence, data space management problems are reduced. Space management in single-entry data sets is simple. New records are positioned randomly. When records are deleted, their space is made available immediately for

new records. In variable entry data sets, record replacement is quite arbitrary. When records are deleted, their freed space is added to a list of available spaces. One advantage of having fixed length records is that any new record may occupy any free space. However, a serious drawback is that some spaces may be wasted in variable entry data sets where record types of different lengths may exist and all of them are required to be of fixed record length.

(10) System Application Flexibility

ADABAS. As ADABAS loads each data record into the data base, it assigns an Internal Sequence Number (INS) to that record. The user can logically identify the record by its ISN. The ISN of a record remains assigned to it until the user deletes it from the data base. The physical block number in which the particular ISN is stored is also noted by the system. This separation of the logical identification of a record from its physical location results in the following advantages:

- Fast retrieval and updating capability
- Complete device independence
- Complete independence of logical data contents from the physical organization of the data.

Furthermore, ADABAS has:

- The ability to define a wide variety of data types and data structures within the data base
- The ability to retrieve data at the field level rather than at the record level; this permits a high degree of data independence between each user application and the manner in which the data is physically organized and maintained
- The ability to establish new fields within existing files, dynamically update non-key fields to key fields, dynamically create or dissolve logical interfile relationships, dynamically expand the overall size of the data base, all without having to recreate, reload, or reorganize the data base.

IDMS. The user of the IDMS may create a set relationship between any two records in the data base regardless of their physical locations. If a designated set relationship requires access to two areas, the system will ensure that both areas are available before access is permitted. Thus, a data element or a selected group of data elements within a record may be available to any number of different applications that require access to it.

The subschema facility of IDMS allows the user to access data items and records that are not logically connected with other data being processed by an application. The subschema would make the necessary data items, records, areas, and sets visible to the invoking program.

IMS. IMS/VS provides both the capability of an application program to reference arbitrarily located units of data independent of their placement within the data base and the capability of interchanging data between two programs or jobs. The user describes his physical and logical data bases through the tools provided by the data base definition facilities of the system. Each application program, in turn, specifies to which of the segments in a logical data base the program is sensitive. The combination of these two facilities allows any application program to be constrianed to a limited subset of the data contained in the data base. It also permits selection of this subset from the set of interrelated data items, regardless of their methods of storage. For example, having described a set of physical data bases and their logical interrelationship, an application program can be limited to only a few elements from one or more of the defined data bases, or through extension of the program control block, it can be granted access to all segments in all data bases. This has the effect of providing a pseudo data base definition process to the object time programs. Although the user can structure his logical and physical data bases in a variety of ways, he must always adhere to the rule that the hierarchy, either logical or physical, must be traced when accessing data segments below the root segment level.

System 2000. System 2000 provides a relatively flexible data base. Data placement is totally transparent to the application program. When seeking to retrieve a specific entry subject to different qualifying criteria, the user has available a variety of techniques for specifying

the data. The interprogram communication capability for transferring data among different jobs and programs is available to the procedural language processing users only. This is a function of the operating system under which it is executing. There are no apparent provisions within System 2000 for performing this function within the confines of the system itself.

TOTAL. TOTAL has a significant degree of flexibility in its application of a data base to problem solving. This degree of generality resides in the single entry record which, although placed by a single randomized key, may carry within it key values for other keys. This means that the single entry record may stand at the head of many lists in a variable entry file. As such, it provides a concentrated key value reference: its first portion is the single entry file key, the second portion is associated with a selected linkage path in the variable entry file.

A TOTAL data base consists of two types of files: single entry and variable entry. The single entry files may be regarded as tabulations to the heads of lists in the variable entry file. As such, the single entry file may link to any variable entry file in the data base. That is, a single tabulation, represented by one single entry file, may represent a tabular indexing to a number of variable entry files in the data base system. Furthermore, variable entry files may have such an indexed reference from any number of single entry files. All of this gives TOTAL an exceptional capability of interconnecting and relating files of the data base system, close to a "true" network capability. Consequently, TOTAL provides flexibility of application of data already in the data base to new problems. Data to be added to the data base for the new problem appears as a new file which may have the variable entry format. The user can then choose to employ an existing single entry file or create one with a special key for referencing the new variable entry file. Furthermore, all data carried in files of the existing data base system can now be made available to any user of the system.

(11) Query Capabilities

ADABAS. In addition to the data management function, ADABAS also offers data query ability via ADASCRIPT (a query language through

which the user can formulate data base commands in a natural language form). This language is oriented toward use by nonprogramming personnel. ADASCRIPT performs all necessary syntax checking and command translation and also returns the formatted results of the operation to the user. This facility is particularly useful when employed in an on-line conversational mode of operation.

IDMS. The only query facility provided in IDMS is through the data management language (DML). DML calls may be coded directly by the user or passed through the DML preprocessor. Data qualification does not exist except through specification of calc keys or through implicit set relationships.

IMS. IMS/2 has three levels of query capabilities:

- The DL/1 and its host language
- The generalized information system/2
- The interactive query facility.

These three levels vary considerably in terms of how they perform their functions and how the user structures his queries. Each offers a different set of facilities and requires skills to effectively utilize those facilities.

In DL/1, the query facilities are represented by the retrieval commands and their associated Segment Search Arguments (SSA's) coupled with the tools of the host language (COBOL, PL/1, or Assembler). At the other two levels, the query facilities are incorporated into a stand-alone procedural language that has its own translation facilities for scanning, passing, and executing command sequences. Limited heuristic searching of a type can be performed as users are provided the ability to create separate, saved files from searches and then process queries on these files.

System 2000. The query capabilities of System 2000 are available through the procedural language interfaces or through the two syntaxes of natural language. Data on which operations are to be performed may be specified by a WHERE clause.

In the procedural language processing, the unit of update or retrieval is limited to only one data set at a time. Update or deletion may occur only after the data set has been retrieved.

In the natural language processing, each command specifies the selection criteria for the data and the operations to be performed. No commands are available to specify operations based on a current position. Hence, a command to update the data base must specify which elements are to be changed, as well as their new contents.

TOTAL. The query capabilities of the various TOTAL configurations are basically incorporated in the Data Management Language (DML) facilities of the package. Query and report generation capabilities will soon be made available. TOTAL can also be set up to interface with other independent query systems, such as CULPRIT and SCORE.

SOCRATES, a CINCOM marketed package is also available, permitting users to access the TOTAL data base and, optionally, sequential files. SOCRATES can automatically extract data from both single entry and variable entry files. It can virtually handle any network relationship defined, thus providing a powerful extraction and reporting capability.

(12) Restrictions and Limits Versus Assets

ADABAS. ADABAS provides comparatively limited query capabilities. Lacking are:

- The ability to let users retrieve records based on the existence or nonexistence of a data field; that is, it fails to distinguish null values from zeros or blanks
- The ability to perform phrase searching; that is, search for multiple words imbedded in a textual data field
- The ability to display the data element dictionary if requested
- The support of a synonym table or controlled vocabulary so that the retriever may look for a class of semantically equivalent terms.

Users of ADABAS find its "blackbox" approach appealing, i.e., users never need to know "how" or "where" their data are stored. Adding and deleting fields can be achieved without unloading and reloading the data base.

IDMS. IDMS does not provide any telecommunications handling facility of its own. The Generalized Communications Interface (GCI) option is not, in itself, a telecommunication facility. Instead, it includes several modules which allow running the system in a central, multi-user environment. The GCI allows IDMS coupling to most telecommunications monitors, including intercom and CICS.

IDMS may be used with any host language (for example, FORTRAN, PL/1, Assembler) that supports a CALL statement. For COBOL, a special set of data manipulation statements is available for data-handling functions.

IMS. IMS is an extremely powerful data-handling system, but the overhead costs are comparatively high. A great deal of administrative support is necessary on a continuing basis; special purpose programs must be written for different applications. Thus, the system itself tends to consume professional resources.

Some users find the calling conventions cumbersome and difficult to learn. Overall, education and training for effective use of IMS is relatively expensive.

System 2000. In terms of overall system complexity, System 2000 falls somewhere between the extremes presented by IMS and TOTAL. System 2000 emphasizes the construction of efficient indexes to data records, based on inversion of individual data fields. Unlike other inverted DBMS's, System 2000 emphasizes hierarchical data structuring, with up to 32 levels in a given tree. Although powerful in a number of respect, some critics have argued that the hierarchical approach of System 2000 does not achieve the data networking possible in other DBMS's.

TOTAL. One of the restrictions is that the record type must always be either a single-entry "master" record or a variable-entry "detailed" record. It cannot be a "master" in one set type and a "detailed" in another. This restriction prohibits that a hierarchy in TOTAL can never be directly expressed with more than two levels. Additionally, TOTAL does not support variable length records.

In conclusion, Table 3 sums up the capabilities and characteristics of the seven data base management systems described in this report.

TABLE 3. SUMMARY OF SEVEN DBMS CAPABILITIES

	ADABAS	IMS/VS	IMS/VS with GIS	IMDS	SYSTEM 2000	TOTAL	BASIS
Numeric Oriented	X	X	X	X	X	X	X
Textual Oriented							X
Retrieval Oriented	X		X		X		X
Update Oriented		X		X		X	
Self-Contained Query	X		X		X		X
Update	X		X		X		X
Report Writing	X		X		X		X
Host Language Interface Query	X	X	X	X	X		X
Update	X	X	X	X	X		X
Report Writing	X	X	X		X		X
Ad-hoc Query	X		X	X	X		
Predetermined Transaction		X			X	X	
Predefined Process via User Language Macro			X		X		X
Predefined Process via Procedural Programming Language	X	X	X	X	X	X	X
Network Data Structure				X		X	
Non Network Data Structure	X	X	X		X		X

d. Documentation of Verification and Validation Requirements

The verification and validation of the conceptual computerized MC/DG depends largely on the input from the potential users of the system. Two major sources of input are the survey of aerospace designers and also the industry briefing on the MC/DG, scheduled on April 12, 1979.

The survey of aerospace industry designers, conducted at the beginning of this contract, was used to determine the necessary form and function of the conceptual computerized MC/DG system. That survey revealed the following designer attitudes towards the computerized MC/DG:

- The MC/DG should be easy and quick to use.
- The MC/DG would be used in all phases of design, from conceptual through detail design.
- The MC/DG should be structured to "guide" the designer through the trade-off process--a feature that would be very beneficial to inexperienced designers.
- The designers considered x-y graphs and text to be the most useful presentation modes for MC/DG information.
- The ability to store, in the computer, discrete parts of an assembly or subassembly would be useful.
- The ability to interact with design and analysis programs in conjunction with the MC/DG was considered valuable.

The industry briefing, in April, 1979, provided the opportunity to verify that the proposed computerized MC/DG system will meet the needs of industry. This will be accomplished by allowing these potential users to see the conceptual system demonstrated, comment on what changes they would like to see made, and provide guidance for future implementation and development of the MC/DG computerized system.

APPENDIX A

THE DEMONSTRATION MC/DG DATA BASE DDL PROGRAM

```
DESCRIPTION(GUIDE):
*/                              TEST MOD OF NOV 12 78
*/
*/    D D L FOR MC/OG DATA BASE
*/
*/    PHASE II
*/
BASIS.DATA.BASE.SECTION;
*/
  FILE.DEFINITIONS:
*/
   HEAD.FILE
         PFN=BASISHEAD1MCDG,ID=MCDG,SN=ISSMP,KEY.TYPE(RANDOM);
              MIN.KEY(1),MAX.KEY(999999999),MAX.RECORD(10000),
              MAX.NR.FIELDS(999),HIGHEST.FIELD.NR(999);
*/
   INDEX.FILE
         PFN=BASISINDX1MCDG,ID=MCDG,SN=ISSMP,KEY.SIZE(55);
              PACKING.FACTOR(2),ACC.NR.SIZE(30);
*/
   RANGE.FILE
         PFN=BASISRANGEMCDG,ID=MCDG,SN=ISSMP,KEY.SIZE(30);
              MAX.PACKING.FACTOR(3);
*/
*/
  OPTIONS:
         ADJACENT.TERMS(6);
         TERMINAL.LINE(72);
         PRINTER.LINE(120);
         PRINT.ROUTINE(COMP);
         EXTRA.COMP(ON);
         COMP.PAGE(ON);
         ACC.NR.FIELD(1);
         RANGE.SEARCH(ON);
         DESCRIBE(FIRST);
         PAGE(VALUE(5.0),DOC.FACTOR(1.0),FIELD.FACTOR(0.0);
         LABEL(ON);FIELD.NUMBER(OFF);
         MONITOR(ON);
         INDENT(09);
         SIGN.ON(ON);SIGN.OFF(ON);
         SECURITY(TYPE(EQ),SIZE(1));
         LINKS(OFF);
         DEFINED.VARIABLES(DEFINITIONS);
*/
  SECURITY.DESCRIPTION:
*/
         ID=≠CLAYDON≠,  PW=≠        ≠,     CODE=
         ID=≠LARSON≠ ,  PW=≠        ≠,     CODE=
         ID=≠FISHER≠ ,  PW=≠        ≠,     CODE=
         ID=≠MOON≠   ,  PW=≠        ≠,     CODE=
         ID=≠MCDG≠,     PW=≠        ≠,     CODE=
         ID=≠BASIS1≠ ,  PW=≠        ≠,     CODE=
```

```
        ID=≠BASIS2≠ , PW=≠      ≠,    CODE= ;
        ID=≠BASIS3≠ , PW=≠      ≠,    CODE= ;
*/
*/
  FIELD.PREFIX,PREFIX.DELIMITER≠:≠;
*/
    PREFIX(001)=≠ACC≠;
    PREFIX(002)=≠GROUPTEC≠;
    PREFIX(003)=≠FILE≠;
    PREFIX(004)=≠SUBFILE≠;
    PREFIX(005)=≠PARTCODE≠;
    PREFIX(006)=≠MATERIAL≠;
    PREFIX(007)=≠MATFINAL≠;
    PREFIX(008)=≠MFGMETH≠;
    PREFIX(009)=≠DATADATE≠;
    PREFIX(010)=≠DATATYPE≠;
    PREFIX(011)=≠LOTSIZE≠;
    PREFIX(012)=≠DESCNUSE≠;
    PREFIX(013)=≠INSTMETH≠;
    PREFIX(016)=≠LENGTH≠;
    PREFIX(017)=≠WIDTH≠;
    PREFIX(018)=≠AREA≠;
    PREFIX(019)=≠HEIGHT≠;
    PREFIX(020)=≠THICK≠;
    PREFIX(021)=≠NJOGGLES≠;
    PREFIX(022)=≠NFLHOLES≠;
    PREFIX(023)=≠NOFPARTS≠;
    PREFIX(024)=≠NFASTENR≠;
    PREFIX(025)=≠CASSCOST≠;
    PREFIX(026)=≠BPCOST≠;
    PREFIX(027)=≠BPNRCOST≠;
    PREFIX(028)=≠LTNRCOST≠;
    PREFIX(029)=≠ETNRCOST≠;
    PREFIX(031)=≠NRTCOST≠;
    PREFIX(032)=≠JOGGLES≠;
    PREFIX(033)=≠FLHOLES≠;
    PREFIX(034)=≠BEADS≠;
    PREFIX(035)=≠HTREAT≠;
    PREFIX(036)=≠SURFIN≠;
    PREFIX(037)=≠TOLERAN≠;
    PREFIX(038)=≠LINTRIM≠;
    PREFIX(039)=≠ENDTRIM≠;
    PREFIX(040)=≠UNFLHOLE≠;
    PREFIX(041)=≠DPCOST≠;
    PREFIX(042)=≠EDGEDBLR≠;
    PREFIX(043)=≠STGRDBLR≠;
    PREFIX(044)=≠PADDBLR≠;
    PREFIX(045)=≠INSHCLPS≠;
    PREFIX(046)=≠FASTYPE1≠;
    PREFIX(047)=≠FASTYPE2≠;
    PREFIX(048)=≠FASTYPE3≠;
    PREFIX(049)=≠FASTYPE4≠;
```

```
PREFIX(050)=≠FASTYPE5≠;
PREFIX(052)=≠BPCOSTC0≠;
PREFIX(053)=≠BPCOSTC1≠;
PREFIX(054)=≠BPCOSTC2≠;
PREFIX(055)=≠BPCOSTC3≠;
PREFIX(056)=≠BPCOSTC4≠;
PREFIX(057)=≠BPCOSTC5≠;
PREFIX(058)=≠BPCOSTC6≠;
PREFIX(059)=≠BPCOSTC7≠;
PREFIX(060)=≠NRTC0≠;
PREFIX(061)=≠NRTC1≠;
PREFIX(062)=≠NRTC2≠;
PREFIX(063)=≠NRTC3≠;
PREFIX(064)=≠NRTC4≠;
PREFIX(065)=≠NRTC5≠;
PREFIX(066)=≠NRTC6≠;
PREFIX(067)=≠NRTC7≠;
PREFIX(070)=≠JOGGLEC0≠;
PREFIX(071)=≠JOGGLEC1≠;
PREFIX(072)=≠JOGGLEC2≠;
PREFIX(073)=≠JOGGLEC3≠;
PREFIX(074)=≠JOGGLEC4≠;
PREFIX(075)=≠JOGGLEC5≠;
PREFIX(076)=≠JOGGLEC6≠;
PREFIX(077)=≠JOGGLEC7≠;
PREFIX(080)=≠FLHOLEC0≠;
PREFIX(081)=≠FLHOLEC1≠;
PREFIX(082)=≠FLHOLEC2≠;
PREFIX(083)=≠FLHOLEC3≠;
PREFIX(084)=≠FLHOLEC4≠;
PREFIX(085)=≠FLHOLEC5≠;
PREFIX(086)=≠FLHOLEC6≠;
PREFIX(087)=≠FLHOLEC7≠;
PREFIX(090)=≠HTREATC0≠;
PREFIX(091)=≠HTREATC1≠;
PREFIX(092)=≠HTREATC2≠;
PREFIX(093)=≠HTREATC3≠;
PREFIX(094)=≠HTREATC4≠;
PREFIX(095)=≠HTREATC5≠;
PREFIX(096)=≠HTREATC6≠;
PREFIX(097)=≠HTREATC7≠;
PREFIX(100)=≠LTRIMC0≠;
PREFIX(101)=≠LTRIMC1≠;
PREFIX(102)=≠LTRIMC2≠;
PREFIX(103)=≠LTRIMC3≠;
PREFIX(104)=≠LTRIMC4≠;
PREFIX(105)=≠LTRIMC5≠;
PREFIX(106)=≠LTRIMC6≠;
PREFIX(107)=≠LTRIMC7≠;
PREFIX(110)=≠ETRIMC0≠;
PREFIX(111)=≠ETRIMC1≠;
PREFIX(112)=≠ETRIMC2≠;
```

```
FIELD(020) = USE.FORMAT(05),LABEL(20),NAME(THICK),UNIT.LABEL(2);
FIELD(021) = USE.FORMAT(04),LABEL(21),NAME(NJOGGLES);
FIELD(022) = USE.FORMAT(04),LABEL(22),NAME(NFLHOLES);
FIELD(023) = USE.FORMAT(04),LABEL(23),NAME(NOFPARTS);
FIELD(024) = USE.FORMAT(04),LABEL(24),NAME(NFASTENR);
FIELD(025) = USE.FORMAT(05),LABEL(25),NAME(CASSCOST),UNIT.LABEL(3);
FIELD(026) = USE.FORMAT(05),LABEL(26),NAME(BPCOST),UNIT.LABEL(3);
FIELD(027) = USE.FORMAT(05),LABEL(27),NAME(BPNRCOST),UNIT.LABEL(3);
FIELD(028) = USE.FORMAT(05),LABEL(28),NAME(LTNRCOST),UNIT.LABEL(3);
FIELD(029) = USE.FORMAT(05),LABEL(29),NAME(ETNRCOST),UNIT.LABEL(3);
FIELD(031) = USE.FORMAT(05),LABEL(31),NAME(NRTCOST),UNIT.LABEL(3);
FIELD(032) = USE.FORMAT(05),LABEL(32),NAME(JOGGLES)  ,UNIT.LABEL(3);
FIELD(033) = USE.FORMAT(05),LABEL(33),NAME(FLHOLES)  ,UNIT.LABEL(3);
FIELD(034) = USE.FORMAT(05),LABEL(34),NAME(BEADS)    ,UNIT.LABEL(3);
FIELD(035) = USE.FORMAT(05),LABEL(35),NAME(HTREAT)   ,UNIT.LABEL(3);
FIELD(036) = USE.FORMAT(05),LABEL(36),NAME(SURFIN)   ,UNIT.LABEL(3);
FIELD(037) = USE.FORMAT(05),LABEL(37),NAME(TOLERAN)  ,UNIT.LABEL(3);
FIELD(038) = USE.FORMAT(05),LABEL(38),NAME(LINTRIM)  ,UNIT.LABEL(3);
FIELD(039) = USE.FORMAT(05),LABEL(39),NAME(ENDTRIM)  ,UNIT.LABEL(3);
FIELD(040) = USE.FORMAT(05),LABEL(40),NAME(UNFLHOLE),UNIT.LABEL(3);
FIELD(041) = USE.FORMAT(05),LABEL(41),NAME(DPCOST)   ,UNIT.LABEL(3);
FIELD(042) = USE.FORMAT(05),LABEL(42),NAME(EDGEDBLR),UNIT.LABEL(3);
FIELD(043) = USE.FORMAT(05),LABEL(43),NAME(STGRDBLR),UNIT.LABEL(3);
FIELD(044) = USE.FORMAT(05),LABEL(44),NAME(PADDBLR),UNIT.LABEL(3);
FIELD(045) = USE.FORMAT(05),LABEL(45),NAME(INSHCLIP),UNIT.LABEL(3);
FIELD(046) = USE.FORMAT(04),LABEL(46),NAME(FASTYPE1);
FIELD(047) = USE.FORMAT(04),LABEL(47),NAME(FASTYPE2);
FIELD(048) = USE.FORMAT(04),LABEL(48),NAME(FASTYPE3);
FIELD(049) = USE.FORMAT(04),LABEL(49),NAME(FASTYPE4);
FIELD(050) = USE.FORMAT(04),LABEL(50),NAME(FASTYPE5);
FIELD(052) = USE.FORMAT(05),LABEL(052),NAME(BPCOSTC0);
FIELD(053) = USE.FORMAT(05),LABEL(053),NAME(BPCOSTC1);
FIELD(054) = USE.FORMAT(05),LABEL(054),NAME(BPCOSTC2);
FIELD(055) = USE.FORMAT(05),LABEL(055),NAME(BPCOSTC3);
FIELD(056) = USE.FORMAT(05),LABEL(056),NAME(BPCOSTC4);
FIELD(057) = USE.FORMAT(05),LABEL(057),NAME(BPCOSTC5);
FIELD(058) = USE.FORMAT(05),LABEL(058),NAME(BPCOSTC6);
FIELD(059) = USE.FORMAT(05),LABEL(059),NAME(BPCOSTC7);
FIELD(060) = USE.FORMAT(05),LABEL(060),NAME(NRTC0);
FIELD(061) = USE.FORMAT(05),LABEL(061),NAME(NRTC1);
FIELD(062) = USE.FORMAT(05),LABEL(062),NAME(NRTC2);
FIELD(063) = USE.FORMAT(05),LABEL(063),NAME(NRTC3);
FIELD(064) = USE.FORMAT(05),LABEL(064),NAME(NRTC4);
FIELD(065) = USE.FORMAT(05),LABEL(065),NAME(NRTC5);
FIELD(066) = USE.FORMAT(05),LABEL(066),NAME(NRTC6);
FIELD(067) = USE.FORMAT(05),LABEL(067),NAME(NRTC7);
FIELD(070) = USE.FORMAT(05),LABEL(070),NAME(JOGGLEC0);
FIELD(071) = USE.FORMAT(05),LABEL(071),NAME(JOGGLEC1);
FIELD(072) = USE.FORMAT(05),LABEL(072),NAME(JOGGLEC2);
FIELD(073) = USE.FORMAT(05),LABEL(073),NAME(JOGGLEC3);
FIELD(074) = USE.FORMAT(05),LABEL(074),NAME(JOGGLEC4);
FIELD(075) = USE.FORMAT(05),LABEL(075),NAME(JOGGLEC5);
```

```
FIELD(076) = USE.FORMAT(05),LABEL(076),NAME(JOGGLEC6);
FIELD(077) = USE.FORMAT(05),LABEL(077),NAME(JOGGLEC7);
FIELD(080) = USE.FORMAT(05),LABEL(080),NAME(FLHOLEC0);
FIELD(081) = USE.FORMAT(05),LABEL(081),NAME(FLHOLEC1);
FIELD(082) = USE.FORMAT(05),LABEL(082),NAME(FLHOLEC2);
FIELD(083) = USE.FORMAT(05),LABEL(083),NAME(FLHOLEC3);
FIELD(084) = USE.FORMAT(05),LABEL(084),NAME(FLHOLEC4);
FIELD(085) = USE.FORMAT(05),LABEL(085),NAME(FLHOLEC5);
FIELD(086) = USE.FORMAT(05),LABEL(086),NAME(FLHOLEC6);
FIELD(087) = USE.FORMAT(05),LABEL(087),NAME(FLHOLEC7);
FIELD(090) = USE.FORMAT(05),LABEL(090),NAME(HTREATC0);
FIELD(091) = USE.FORMAT(05),LABEL(091),NAME(HTREATC1);
FIELD(092) = USE.FORMAT(05),LABEL(092),NAME(HTREATC2);
FIELD(093) = USE.FORMAT(05),LABEL(093),NAME(HTREATC3);
FIELD(094) = USE.FORMAT(05),LABEL(094),NAME(HTREATC4);
FIELD(095) = USE.FORMAT(05),LABEL(095),NAME(HTREATC5);
FIELD(096) = USE.FORMAT(05),LABEL(096),NAME(HTREATC6);
FIELD(097) = USE.FORMAT(05),LABEL(097),NAME(HTREATC7);
FIELD(100) = USE.FORMAT(05),LABEL(100),NAME(LTRIMC0);
FIELD(101) = USE.FORMAT(05),LABEL(101),NAME(LTRIMC1);
FIELD(102) = USE.FORMAT(05),LABEL(102),NAME(LTRIMC2);
FIELD(103) = USE.FORMAT(05),LABEL(103),NAME(LTRIMC3);
FIELD(104) = USE.FORMAT(05),LABEL(104),NAME(LTRIMC4);
FIELD(105) = USE.FORMAT(05),LABEL(105),NAME(LTRIMC5);
FIELD(106) = USE.FORMAT(05),LABEL(106),NAME(LTRIMC6);
FIELD(107) = USE.FORMAT(05),LABEL(107),NAME(LTRIMC7);
FIELD(110) = USE.FORMAT(05),LABEL(110),NAME(ETRIMC0);
FIELD(111) = USE.FORMAT(05),LABEL(111),NAME(ETRIMC1);
FIELD(112) = USE.FORMAT(05),LABEL(112),NAME(ETRIMC2);
FIELD(113) = USE.FORMAT(05),LABEL(113),NAME(ETRIMC3);
FIELD(114) = USE.FORMAT(05),LABEL(114),NAME(ETRIMC4);
FIELD(115) = USE.FORMAT(05),LABEL(115),NAME(ETRIMC5);
FIELD(116) = USE.FORMAT(05),LABEL(116),NAME(ETRIMC6);
FIELD(117) = USE.FORMAT(05),LABEL(117),NAME(ETRIMC7);
FIELD(120) = USE.FORMAT(05),LABEL(120),NAME(UFHOLEC0);
FIELD(121) = USE.FORMAT(05),LABEL(121),NAME(UFHOLEC1);
FIELD(122) = USE.FORMAT(05),LABEL(122),NAME(UFHOLEC2);
FIELD(123) = USE.FORMAT(05),LABEL(123),NAME(UFHOLEC3);
FIELD(124) = USE.FORMAT(05),LABEL(124),NAME(UFHOLEC4);
FIELD(125) = USE.FORMAT(05),LABEL(125),NAME(UFHOLEC5);
FIELD(126) = USE.FORMAT(05),LABEL(126),NAME(UFHOLEC6);
FIELD(127) = USE.FORMAT(05),LABEL(127),NAME(UFHOLEC7);
FIELD(130) = USE.FORMAT(04),LABEL(130),NAME(PLY0);
FIELD(131) = USE.FORMAT(04),LABEL(131),NAME(PLY45);
FIELD(132) = USE.FORMAT(04),LABEL(132),NAME(PLY90);
FIELD(133) = USE.FORMAT(04),LABEL(133),NAME(STRIP0);
FIELD(134) = USE.FORMAT(04),LABEL(134),NAME(STRIP45);
FIELD(135) = USE.FORMAT(04),LABEL(135),NAME(STRIP90);
FIELD(136) = USE.FORMAT(04),LABEL(136),NAME(STRINGER);
FIELD(137) = USE.FORMAT(04),LABEL(137),NAME(FRAMES);
```

MAPPING SECTION

```
*/
   MAP(151)=NAME(A),SEQUENCE(1,4,5,6);
   MAP(152)=NAME(B),SEQUENCE(5,6,8,11);
   MAP(153)=NAME(C),SEQUENCE(8,11,31);
   MAP(155)=NAME(E),SEQUENCE(5,6,8,11,31);
*/
*/
*/
  LABELS:
*/
   LABEL(001) = ≠ACCESSION NUMBER             ≠;
   LABEL(002) = ≠GROUP TECHNOLOGY             ≠;
   LABEL(003) = ≠FILE NAME                    ≠;
   LABEL(004) = ≠SUBFILE NAME                 ≠;
   LABEL(005) = ≠DISCRETE PART CODE           ≠;
   LABEL(006) = ≠MATERIAL USED                ≠;
   LABEL(007) = ≠MATERIAL FINAL CONDITION     ≠;
   LABEL(008) = ≠MANUFACTURING METHOD         ≠;
   LABEL(009) = ≠DATE DATA GENERATED          ≠;
   LABEL(010) = ≠TYPE OF DATA SUBFILE         ≠;
   LABEL(011) = ≠MANUFACTURING LOT SIZE       ≠;
   LABEL(012) = ≠BRIEF PART DESCRIPTION       ≠;
   LABEL(013) = ≠INSTALLATION METHOD          ≠;
   LABEL(016) = ≠PART MEASUREMENT 1,LENGTH    ≠;
   LABEL(017) = ≠PART MEASUREMENT 2, WIDTH    ≠;
   LABEL(018) = ≠PART MEASUREMENT 3, AREA     ≠;
   LABEL(019) = ≠PART MEASUREMENT 4, HEIGHT   ≠;
   LABEL(020) = ≠PART MEASUREMENT 5, THICKNESS≠;
   LABEL(021) = ≠NUMBER OF JOGGLES            ≠;
   LABEL(022) = ≠NUMBER OF FLANGED HOLES      ≠;
   LABEL(023) = ≠NUMBER OF PARTS              ≠;
   LABEL(024) = ≠NUMBER OF FASTENERS          ≠;
   LABEL(025) = ≠COMPLETE ASSEMBLY COST       ≠;
   LABEL(026) = ≠BASE PART COST               ≠;
   LABEL(027) = ≠BASE PART NON-RECUR TOOL COST≠;
   LABEL(028) = ≠LIN. TRIM NON-RECUR TOOL COST≠;
   LABEL(029) = ≠END TRIM  NON-RECUR TOOL COST≠;
   LABEL(031) = ≠TOTAL NON-RECURRING TOOL COST≠;
   LABEL(032) = ≠JOGGLES COST                 ≠;
   LABEL(033) = ≠FLANGED HOLES COST           ≠;
   LABEL(034) = ≠BEADS COST                   ≠;
   LABEL(035) = ≠HEAT TREATMENT COST          ≠;
   LABEL(036) = ≠SURFACE FINISH COST          ≠;
   LABEL(037) = ≠TOLERANCE COST               ≠;
   LABEL(038) = ≠LINEAL TRIM COST             ≠;
   LABEL(039) = ≠END TRIM COST                ≠;
   LABEL(040) = ≠CUT-OUTS WITHOUT FLANGES COST≠;
   LABEL(041) = ≠DISCRETE PART COST           ≠;
   LABEL(042) = ≠EDGE DBLR COST               ≠;
   LABEL(043) = ≠STGR DBLR COST               ≠;
   LABEL(044) = ≠PAD  DBLR COST               ≠;
   LABEL(045) = ≠INT.SHR CLIPS COST           ≠;
```

```
LABEL(046) = ≠COUNTERSINK RIVET COST        ≠;
LABEL(047) = ≠UNIVERSAL HEAD RIVET COST     ≠;
LABEL(048) = ≠COUNTERSINK HI-LOK BOLT COST ≠;
LABEL(049) = ≠UNIVERSAL HI-LOK BOLT COST    ≠;
LABEL(050) = ≠TYPE 5 FASTENER COST          ≠;
LABEL(052) = ≠BASE PART COST REG. COEF. C0 ≠;
LABEL(053) = ≠BASE PART COST REG. COEF. C1 ≠;
LABEL(054) = ≠BASE PART COST REG. COEF. C2 ≠;
LABEL(055) = ≠BASE PART COST REG. COEF. C3 ≠;
LABEL(056) = ≠BASE PART COST REG. COEF. C4 ≠;
LABEL(057) = ≠BASE PART COST REG. COEF. C5 ≠;
LABEL(058) = ≠BASE PART COST REG. COEF. C6 ≠;
LABEL(059) = ≠BASE PART COST REG. COEF. C7 ≠;
LABEL(060) = ≠NON-REC TOOL COST REG COEF C0≠;
LABEL(061) = ≠NON-REC TOOL COST REG COEF C1≠;
LABEL(062) = ≠NON-REC TOOL COST REG COEF C2≠;
LABEL(063) = ≠NON-REC TOOL COST REG COEF C3≠;
LABEL(064) = ≠NON-REC TOOL COST REG COEF C4≠;
LABEL(065) = ≠NON-REC TOOL COST REG COEF C5≠;
LABEL(066) = ≠NON-REC TOOL COST REG COEF C6≠;
LABEL(067) = ≠NON-REC TOOL COST REG COEF C7≠;
LABEL(070) = ≠JOGGLES COST REG COEF C0      ≠;
LABEL(071) = ≠JOGGLES COST REG COEF C1      ≠;
LABEL(072) = ≠JOGGLES COST REG COEF C2      ≠;
LABEL(073) = ≠JOGGLES COST REG COEF C3      ≠;
LABEL(074) = ≠JOGGLES COST REG COEF C4      ≠;
LABEL(075) = ≠JOGGLES COST REG COEF C5      ≠;
LABEL(076) = ≠JOGGLES COST REG COEF C6      ≠;
LABEL(077) = ≠JOGGLES COST REG COEF C7      ≠;
LABEL(080) = ≠FLANGED HOLE COST REG COEF C0≠;
LABEL(081) = ≠FLANGED HOLE COST REG COEF C1≠;
LABEL(082) = ≠FLANGED HOLE COST REG COEF C2≠;
LABEL(083) = ≠FLANGED HOLE COST REG COEF C3≠;
LABEL(084) = ≠FLANGED HOLE COST REG COEF C4≠;
LABEL(085) = ≠FLANGED HOLE COST REG COEF C5≠;
LABEL(086) = ≠FLANGED HOLE COST REG COEF C6≠;
LABEL(087) = ≠FLANGED HOLE COST REG COEF C7≠;
LABEL(090) = ≠HEAT TREAT COST REG COEF C0   ≠;
LABEL(091) = ≠HEAT TREAT COST REG COEF C1   ≠;
LABEL(092) = ≠HEAT TREAT COST REG COEF C2   ≠;
LABEL(093) = ≠HEAT TREAT COST REG COEF C3   ≠;
LABEL(094) = ≠HEAT TREAT COST REG COEF C4   ≠;
LABEL(095) = ≠HEAT TREAT COST REG COEF C5   ≠;
LABEL(096) = ≠HEAT TREAT COST REG COEF C6   ≠;
LABEL(097) = ≠HEAT TREAT COST REG COEF C7   ≠;
LABEL(100) = ≠LINEAL TRIM COST REG COEF C0 ≠;
LABEL(101) = ≠LINEAL TRIM COST REG COEF C1 ≠;
LABEL(102) = ≠LINEAL TRIM COST REG COEF C2 ≠;
LABEL(103) = ≠LINEAL TRIM COST REG COEF C3 ≠;
LABEL(104) = ≠LINEAL TRIM COST REG COEF C4 ≠;
LABEL(105) = ≠LINEAL TRIM COST REG COEF C5 ≠;
LABEL(106) = ≠LINEAL TRIM COST REG COEF C6 ≠;
```

```
     LABEL(107) = ≠LINEAL TRIM COST REG COEF C7 ≠;
     LABEL(110) = ≠END TRIM COST REG COEF C0    ≠;
     LABEL(111) = ≠END TRIM COST REG COEF C1    ≠;
     LABEL(112) = ≠END TRIM COST REG COEF C2    ≠;
     LABEL(113) = ≠END TRIM COST REG COEF C3    ≠;
     LABEL(114) = ≠END TRIM COST REG COEF C4    ≠;
     LABEL(115) = ≠END TRIM COST REG COEF C5    ≠;
     LABEL(116) = ≠END TRIM COST REG COEF C6    ≠;
     LABEL(117) = ≠END TRIM COST REG COEF C7    ≠;
     LABEL(120) = ≠UNFLANGED HOLES COST REG C0  ≠;
     LABEL(121) = ≠UNFLANGED HOLES COST REG C1  ≠;
     LABEL(122) = ≠UNFLANGED HOLES COST REG C2  ≠;
     LABEL(123) = ≠UNFLANGED HOLES COST REG C3  ≠;
     LABEL(124) = ≠UNFLANGED HOLES COST REG C4  ≠;
     LABEL(125) = ≠UNFLANGED HOLES COST REG C5  ≠;
     LABEL(126) = ≠UNFLANGED HOLES COST REG C6  ≠;
     LABEL(127) = ≠UNFLANGED HOLES COST REG C7  ≠;
     LABEL(130) = ≠PLY COUNT 0 DEGREE           ≠;
     LABEL(131) = ≠PLY COUNT 45 DEGREES         ≠;
     LABEL(132) = ≠PLY COUNT 90 DEGREES         ≠;
     LABEL(133) = ≠STRIP COUNT 0 DEGREE         ≠;
     LABEL(134) = ≠STRIP COUNT 45 DEGREES       ≠;
     LABEL(135) = ≠STRIP COUNT 90 DEGREES       ≠;
     LABEL(136) = ≠HAT STRINGERS                ≠;
     LABEL(137) = ≠FRAMES                       ≠;
*/
*/
*/   UNIT LABELS FOR PART DIMENSIONS AND COSTS
*/
     UNIT.LABEL(1)=≠DEGREES≠;
     UNIT.LABEL(2)=≠INCHES≠;
     UNIT.LABEL(3)=≠MANHRS≠;
     UNIT.LABEL(4)=≠SQ. INS.≠;
*/
  MESSAGES:
*/
  NUMBER(6007)
  TEXT(//(THE MANUFACTURE COST/DESIGN GUIDE SYSTEM));
  NUMBER(6008)
  TEXT(//(LAST UPDATE    >XXXXXXXX>  ));
  NUMBER(6009)
  TEXT(//(TOTAL ITEMS IN DATA BASE=   >9999999>  ));
  NUMBER(6010)
  TEXT(//(MC/DG RETURNS YOU TO INTERCOM));
*/
*/
  DATA.RANGES,PREFIX.DELIMITER(:),SEPARATOR(/);
*/
  FORMAT(1),CLASS(INTEGER),PACKING.FACTOR(2),FIELD.SIZE(30),
  RANGE FROM  100000000  TO  199999999
        FROM  200000000  TO  299999999
        FROM  300000000  TO  399999999
```

```
        FROM   400000000   TO   499999999
        FROM   500000000   TO   599999999
        FROM   600000000   TO   699999999
        FROM   700000000   TO   799999999
        FROM   800000000   TO   899999999
        FROM   900000000   TO   999999999;
  FORMAT(2),CLASS(REAL),PACKING.FACTOR(3),FIELD.SIZE(20),MULT.FACTOR(3),
  RANGE FROM    -100.000   TO     0.000
        FROM       0.001   TO    10.000   BY   0.500
        FROM      10.001   TO   100.000 ;
*/
  FORMAT(3),CLASS(INTEGER),PACKING.FACTOR(3),FIELD.SIZE(20),
  RANGE FROM    -500 TO 0
        FROM       1 TO 2
        FROM       3 TO 4
        FROM       5 TO 24   BY   5
        FROM      50 TO 500
        FROM     501 TO 999999;
*/
  FORMAT(4),CLASS(INTEGER),PACKING.FACTOR(3),FIELD.SIZE(20),
  RANGE FROM      0   TO   4095   BY   512;
*/
  FORMAT(5),CLASS(REAL),PACKING.FACTOR(3),FIELD.SIZE(20),MULT.FACTOR(3),
  RANGE FROM  -500.000   TO      0.000
        FROM     0.001   TO      6.000
        FROM     6.001   TO     12.000
        FROM    12.001   TO     24.000
        FROM    24.001   TO     36.000
        FROM    36.001   TO     48.000
        FROM    48.001   TO     60.000
        FROM    60.001   TO     72.000
        FROM    72.001   TO     96.000
        FROM    96.001   TO    120.000
        FROM   120.001   TO    144.000
        FROM   144.001   TO    240.000
        FROM   240.001   TO    480.000
        FROM   480.001   TO    524.000
        FROM   524.001   TO   1000.000
        FROM  1000.001   TO   3000.000
        FROM  3000.001   TO   5000.000
        FROM  5000.001   TO  99999.000;
*/
  FORMAT(6),CLASS(REAL),PACKING.FACTOR(3),FIELD.SIZE(20),MULT.FACTOR(3),
  RANGE FROM   -48.000   TO     0.000
        FROM     0.001   TO     0.100   BY    0.010
        FROM     0.101   TO    12.100   BY    3.000
        FROM    12.101   TO    96.100   BY   12.000
        FROM    96.101   TO   396.100   BY  150.000
        FROM   396.101   TO  9999.999;
*/
  FORMAT(7),CLASS(REAL),PACKING.FACTOR(3),FIELD.SIZE(20),MULT.FACTOR(3),
  RANGE FROM  -100.000   TO      0.000
```

```
        FROM      0.001   TO      5.000   BY   1.000
        FROM      5.001   TO     20.000   BY   5.000
        FROM     20.001   TO'   100.000   BY  20.000
        FROM    100.001   TO   2000.000   BY 100.00
        FROM   2000.001   TO  99999.99;
*/
  FORMAT(8),CLASS(INTEGER),PACKING.FACTOR(2),FIELD.SIZE(30),
  RANGE FROM   780600 TO 780699
        FROM   780700 TO 780799
        FROM   780800 TO 780899
        FROM   780900 TO 780999
        FROM   781000 TO 781099
        FROM   781100 TO 781199
        FROM   781200 TO 781299
        FROM   790100 TO 791212;
  FORMAT(9),CLASS(INTEGER),PACKING.FACTOR(3),FIELD.SIZE(20),
  RANGE FROM -500 TO 0
        FROM    1 TO 2
        FROM    3 TO 4
        FROM    5 TO 24 BY 5
        FROM   50 TO 500 ;
*/
  PREFIX(ACC)       ,USE.FORMAT(1);
  PREFIX(LOTSIZE)   ,USE.FORMAT(9);
  PREFIX(DATADATE),USE.FORMAT(8);
  PREFIX(LENGTH)    ,USE.FORMAT(5);
  PREFIX(WIDTH)     ,USE.FORMAT(6);
  PREFIX(AREA)      ,USE.FORMAT(5);
  PREFIX(THICK)     ,USE.FORMAT(6);
  PREFIX(HEIGHT)    ,USE.FORMAT(6);
  PREFIX(NJOGGLES),USE.FORMAT(9);
  PREFIX(NFLHOLES),USE.FORMAT(9);
  PREFIX(NOFPARTS),USE.FORMAT(3);
  PREFIX(NFASTENR),USE.FORMAT(3);
  PREFIX(CASSCOST),USE.FORMAT(7);
  PREFIX(BPCOST)    ,USE.FORMAT(7);
  PREFIX(DPCOST)    ,USE.FORMAT(7);
  PREFIX(BPNRCOST),USE.FORMAT(2);
  PREFIX(ETNRCOST),USE.FORMAT(2);
  PREFIX(LTNRCOST),USE.FORMAT(2);
  PREFIX(NRTCOST)   ,USE.FORMAT(7);
  PREFIX(JOGGLES)   ,USE.FORMAT(2);
  PREFIX(FLHOLES)   ,USE.FORMAT(2);
  PREFIX(BEADS)     ,USE.FORMAT(2);
  PREFIX(HTREAT)    ,USE.FORMAT(2);
  PREFIX(SURFIN)    ,USE.FORMAT(2);
  PREFIX(TOLERAN)   ,USE.FORMAT(2);
  PREFIX(LINTRIM)   ,USE.FORMAT(2);
  PREFIX(ENDTRIM)   ,USE.FORMAT(2);
  PREFIX(UNFLHCLE),USE.FORMAT(2);
  PREFIX(EDGEDBLR),USE.FORMAT(2);
  PREFIX(STGRDBLR),USE.FORMAT(2);
```

```
PREFIX(PADDBLR) ,USE.FORMAT(2);
PREFIX(INSHCLPS),USE.FORMAT(2);
PREFIX(FASTYPE1),USE.FORMAT(3);
PREFIX(FASTYPE2),USE.FORMAT(3);
PREFIX(FASTYPE3),USE.FORMAT(3);
PREFIX(FASTYPE4),USE.FORMAT(3);
PREFIX(FASTYPE5),USE.FORMAT(3);
PREFIX(PLY0)     ,USE.FORMAT(3);
PREFIX(PLY45)    ,USE.FORMAT(3);
PREFIX(PLY90)    ,USE.FORMAT(3);
PREFIX(STRIP0)   ,USE.FORMAT(3);
PREFIX(STRIP45)  ,USE.FORMAT(3);
PREFIX(STRIP90)  ,USE.FORMAT(3);
PREFIX(STRINGER),USE.FORMAT(3);
PREFIX(FRAMES)   ,USE.FORMAT(3);
*/
INPUT.DESCRIPTION.SECTION;
*/
INDEXING.DEFINITIONS:
  PREFIX.DELIMITER=≠:≠;
  FIELD(001),RANGE;
  FIELD(003),INDEX,PREFIX=≠FILE≠;
  FIELD(004),INDEX,PREFIX=≠SUBFILE≠;
  FIELD(005),INDEX,PREFIX=≠PARTCODE≠;
  FIELD(006),INDEX,PREFIX=≠MATERIAL≠,OPTION=FREE.TEXT;
  FIELD(006),INDEX,PREFIX=≠MATERIAL≠;
  FIELD(007),INDEX,PREFIX=≠MATFINAL≠;
  FIELD(008),INDEX,PREFIX=≠MFGMETH≠,OPTION=FREE.TEXT;
  FIELD(008),INDEX,PREFIX=≠MFGMETH≠;
  FIELD(009),RANGE,OPTION=DATE.CONVERT;
  FIELD(011),RANGE;
  FIELD(012),INDEX,PREFIX=≠PARTFORM≠,
             OPTION=SUB.FIELD WITH ELEMENT=≠1≠;
  FIELD(012),INDEX,PREFIX=≠USAGE≠,
             OPTION=SUB.FIELD WITH ELEMENT=≠2≠;
  FIELD(012),INDEX,PREFIX=≠DESCRIBE≠,OPTION=FREE.TEXT;
  FIELD(013),INDEX,PREFIX=≠INSTMETH≠,OPTION=FREE.TEXT;
  FIELD(016),RANGE;
  FIELD(017),RANGE;
  FIELD(018),RANGE;
  FIELD(019),RANGE;
  FIELD(020),RANGE;
  FIELD(021),RANGE;
  FIELD(022),RANGE;
  FIELD(023),RANGE;
  FIELD(024),RANGE;
  FIELD(025),RANGE;
  FIELD(026),RANGE;
  FIELD(027),RANGE;
  FIELD(028),RANGE;
  FIELD(029),RANGE;
  FIELD(031),RANGE;
```

```
   FIELD(032),RANGE:
   FIELD(033),RANGE;
   FIELD(034),RANGE:
   FIELD(035),RANGE:
   FIELD(036),RANGE:
   FIELD(037),RANGE:
   FIELD(038),RANGE;
   FIELD(039),RANGE:
   FIELD(040),RANGE:
   FIELD(041),RANGE;
   FIELD(042),RANGE;
   FIELD(043),RANGE;
   FIELD(044),RANGE:
   FIELD(045),RANGE:
   FIELD(046),RANGE;
   FIELD(047),RANGE;
   FIELD(048),RANGE:
   FIELD(049),RANGE:
   FIELD(050),RANGE;
   FIELD(130),RANGE:
   FIELD(131),RANGE:
   FIELD(132),RANGE;
   FIELD(133),RANGE;
   FIELD(134),RANGE:
   FIELD(135),RANGE:
   FIELD(136),RANGE:
   FIELD(137),RANGE;
STOP:
```

APPENDIX B

BASIS FILE MAINTENANCE PROCEDURES

APPENDIX B

BASIS FILE MAINTENANCE PROCEDURES

The data base administrator needs to perform the functions of creating, editing, adding, and removing data for the MC/DG data base. For the concept validation system, these functions are supported by the BASIS file maintenance system.

The BASIS File Maintenance System (FMS) incorporates a set of programs or modules known as the FILE MANAGERS, which simplify translating information from its original form, into a data base usable by BASIS. This system provides the user with a set of software modules that maintain various files--some on tape, and some on direct-access devices used in creating, updating, searching, and retrieving information that is part of the data base. Associated with each file on a BASIS data base is a file manager. This program is the only one that should be used to modify the file it is designated to manage.

The entire file maintenance system is based on a sophisticated transaction processing scheme. Each time a modification to one of the files in the data base is required, a transaction describing the requested change is created. The Source Data File Manager then processes the transaction by taking appropriate actions to modify the data base. This method of file updating insures a high degree of file security and integrity.

In addition, the transaction processing scheme allows for very general indexing of the documents in a data base. Any number of index transactions may be produced for each document. The application programmer decides exactly how each field is to be indexed (i.e., complete inversion, free text, vocabulary control, etc.) and how each index is to be written.

The file managers always print statistics summarizing transactions conducted during execution of a file update. At the user's request, the FMS will also provide an accurate, thorough, audit trail with detailed lists of each modification to the data base. Trace-back methods give the user the ability to regenerate the original data base in its entirety before any updating transactions.

Under certain circumstances, the file managers generate error messages when a user attempts to update a file improperly.

1. DESCRIPTION OF FMS MODULAR PROGRAMS

(a) Primary FMS Modules

The primary modules of the BASIS file maintenance system (FMS) are:

(1) Source Data File Manager (SDFMGR) - Creates and updates the source file of data records. Each new or modified record is provided to this module by a transaction file.

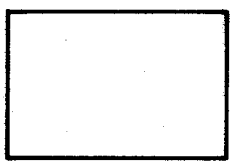
(2) Inverted Index File Manager (IFMGR) - Within the data base, the IFMGR maintains the inverted index used in searching. For each index term in the IFMGR, there is a list of accession numbers for source records containing the index term.

(b) Range Term Processing

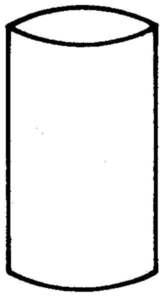
The RANGE VALUE CONTROL (RVCTRL) governs all transactions from fields indexed for range searching by determining the range term under which the field's value should be posted. It creates range term transactions for the index file and range file managers.

The RANGE FILE MANAGER (RFMGR) maintains the range value file of the data base used in true range searching (i.e., ranges defined by the user, not ranges preset by the system). It processes the range term transactions and updates the data value records associated with each range term.

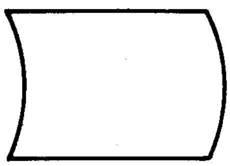
(c) Symbol Explanation



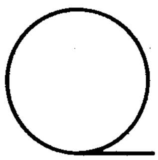
PROCESSOR - This represents a file manager or some other program which will process certain transaction files, and make appropriate changes in other files.



PERMAMENT RMS (Rotating Mass Storage) FILE - These files are normally permamently kept on disk as part of a BASIS data base. (Often these are loaded from tape to disk, they are random access files). Before updating these files, they should be put on tape as backup.



TEMPORARY RMS FILE - These files are kept on disk during the processing steps. When the file maintenance procedure is complete, these files are no longer needed.



BACKUP RMS FILE - These files are kept on disk during the processing steps. When the entire procedure is complete, these files should be copied to TAPE as backup to the data base.

(d) Execution of File Maintenance Programs

In this section, the function and use of each file maintenance program is explained in detail. A flow diagram showing where the program fits into the overall file maintenance procedure is included.

(e) Source Data File Manager

The function of the SDFMGR is to create and update the source (information) file. It does this by reading the source data file transactions produced by the pre-processor program and either inserts, deletes, or replaces source records, depending upon what type of action is indicated by the user. It provides a quality control on the information which is placed in the source file and, as such, should be the only program which creates or updates this file.

The source data transaction file is a binary sequential file. Each transaction record consists of two header words, followed by the actual source file record. The first header word contains the accession number in binary, and the second header word contains the length in characters of the source file record. SDFMGR reads each transaction

and, depending on whether or not a source record exists in the information file with the same accession number, either adds or replaces the record in the file. Note that if SDFMGR reads a transaction which consists of only the accession number header word, it will attempt to delete the record from the source file with the same accession number.

The source file is a Numeric Keyed (NK) file, the key to each record being the accession number for that record. The actual management of the records within the file is handled by the NK library routines. Note that by the use of these routines, any record can be accessed, inserted, or replaced via the key either in random or sequential order. SDFMGR determines the actual file parameters for the source file by retrieving the data description table from the table file. Constants which are used to create and update the source file are contained in this table.

The BASIS data records (or head file records) are all variable length records, with a user-declared maximum record size. Each data element in the record may be a variable length string of characters. No leading or trailing blanks are stored. A single data element may be as large as the record. If certain data elements are missing from some records, this fact is very efficiently stored in the data record (it requires 1 bit). New data elements may be defined and added to new records <u>without</u> changing already existing records. Figure 33 illustrates how the data file manager relates to the source data file and the Table File.

(f) Inverted Index File Manager

The function of IFMGR is to maintain the inverted index file. It performs this function by either creating or updating the inverted index file using the sorted index transactions as input. Since much of what IFMGR does can only be understood with knowledge of the index file structure, a short discussion regarding this point is given below.

The inverted index file is a machine analog of the card catalog used in most libraries. Its primary function is to provide a ready access to information via the use of keywords or keyword phrases. For example, in a library, if a person wanted to find information pertaining to a particular subject, he would go to the card catalog and attempt to find the card or cards which have keywords that best describe the subject area of interest. If found, these cards would have directives on them which would

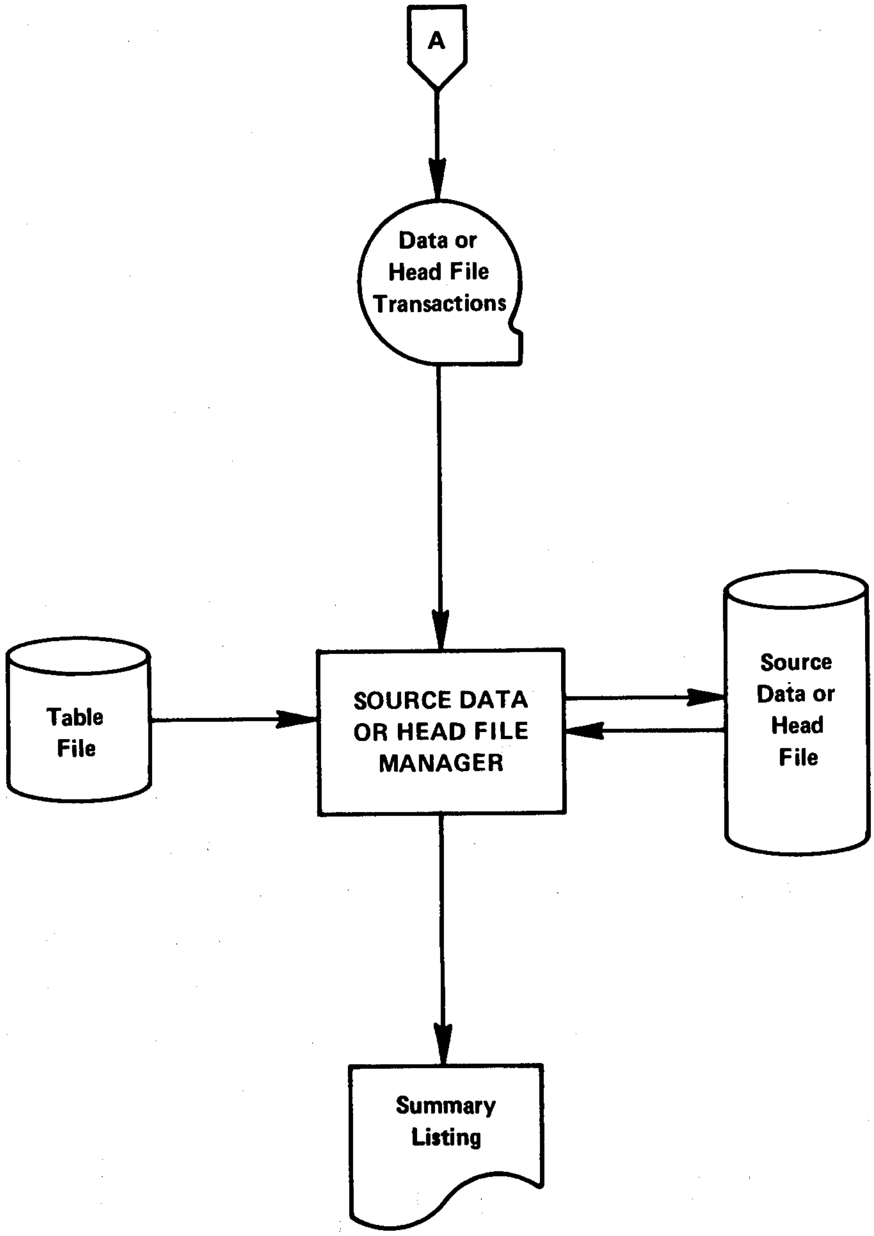
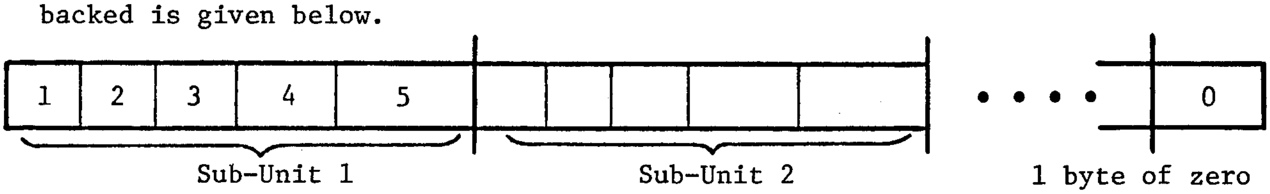


FIGURE 33. HEAD FILE MANAGER

guide the person to where the information is actually stored. The card catalogs are usually maintained with keywords in alphabetical order, allowing the user to have ready access to the subject areas of interest. In the same manner, the inverted index file is so structured that it simulates a card catalog. The file organization used is that of the symbolic keyed (SK) indexed sequential file, with the keyword or keyword phrases being the keys to records which contain accession numbers. These accession numbers, in turn, are keys to the information file (Source Data File). All of the information records which are indexed by a particular keyword have their accession numbers stored as elements of the symbolic keyed file records. The SK file structure allows the user to access, insert, replace, or delete records either in a sequential or random mode of operation.

To be exact, it should be pointed out that the elements of an inverted index file record are actually postings. A posting is composed of an accession number and link, with the link being concatenated to the accession number when stored. If no links are used for a particular data base, then the posting and accession number are the same entity.

The record size of the inverted index file is fixed at a size of 502 computer words. The first word(s) of each record contains header information relating to the postings stored in the record. The rest of the record is devoted to the actual postings. The postings are stored according to a packing factor which the user specifies in the File Description Table (FDT). This packing factor essentially identifies how many postings can be packed into a single computer word. This factor is mainly dependent on how large the postings can become and how many bits are available in a computer word. (There are 60 bits in a CDC 6600 computer word.) A schematic diagram of how several postings might be backed is given below.



Note that within a computer word, the postings are stored right-justified in reverse order. The postings are ordered for a given key, in numerically-increasing order. This is necessary in order to optimize logic operations between different keys.

Since for most applications, some keywords will have a larger number of postings than will fit into one index record, a method has been devised to handle this problem. This scheme amounts to adding a record-sequence indicator as the last two characters of each symbolic key. This indicator would allow a given keyword to have many index records associated with it. The method adopted uses a two-letter sequencing technique, the first record for a keyword having the letters AA appended to it. The second record would have AB appended, etc. In this manner, a given keyword can have a maximum of 26 x 26 = 676 index blocks associated with it. In addition, when the records are inserted into the SK file, the keys to each record will be alphabetically adjacent in the correct order for accessing sequentially. Thus, to find all the postings for a given keyword, the user would perform a random access on the first block and then perform sequential accesses thereafter, until the last block is reached.

Inverted File Manager is the program which builds and maintains the inverted index file. Through the use of the index transactions, the user can direct IFMGR to add or delete postings for a given keyword. For maximum efficiency, the user should sort the index transactions to the following specifications:

Primary Key	-	Term
Secondary Key	-	Accession Number
Tertiary Key	-	Action Flag (deletes should sort before adds).

The sorted transactions should then be used when initiating IFMGR. Figure 34 shows that the index file manager processes the sorted index transactions to update the inverted index file.

(g) Range Term Processing

A powerful feature of BASIS is the capability for numeric range searching. This allows one to request the retrieval of records where one of the numeric data elements must have a value within limits specified by the user. This feature is supported by using an inverted range value file called the range file together with the inverted index file. The application programmer defines the preset groups by which the actual data values

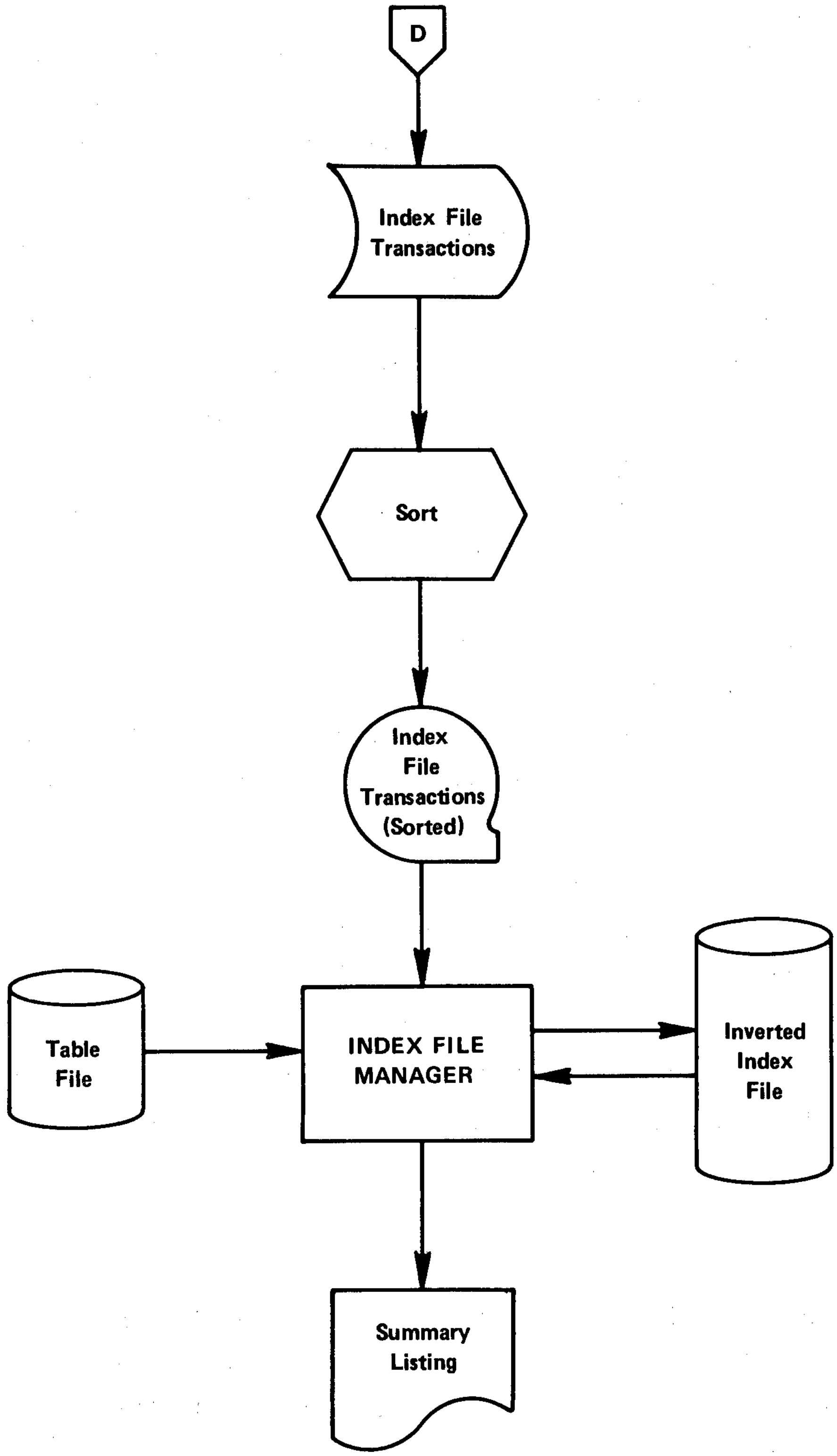


FIGURE 34. INDEX FILE MANAGER

are to be indexed in the DDL. Range term transactions are produced just like index term transactions, except the term type is R and the term has a field prefix, followed by the true data value for the indicated field and accession number. These transactions are first processed by Range Value Control (RVCTRL) to determine what preset range group the indicated term belongs to. Then, the transactions are processed by the Range File Manager (RFMGR) which updates the index and range files. For each preset range term, there is one set of index posting records in the inverted index file, and one set of range value records in the range file. A one-to-one correspondence of accession number and range value is maintained. Quite often, BASIS can satisfy a user's retrieval request on a ranged field by only using the index file. When necessary, both files are used together to determine what records contain range values that are within the users specified limits. This method provides for very quick responses to numeric-range retrieval requests.

Figure 35 shows the Range File Manager, reads the sorted range file transactions, and updates the range file and the inverted index file, simultaneously.

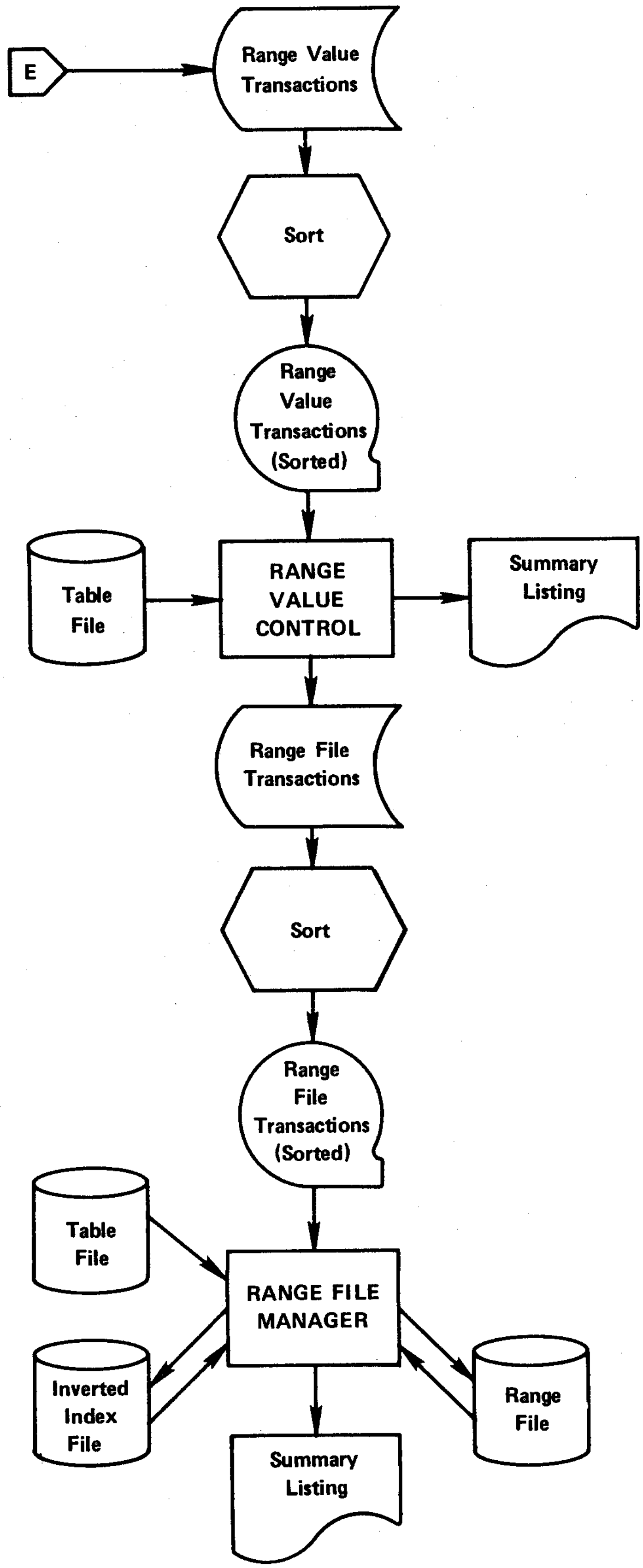


FIGURE 35. RANGE FILE MANAGER

APPENDIX C

INTERACTIVE GRAPHIC DISPLAY MODEL OF THE COMPUTERIZED MC/DG

APPENDIX C

INTERACTIVE GRAPHIC DISPLAY MODEL OF THE COMPUTERIZED MC/DG

1. FUNCTION

An interface between BASIS and the Integrated Graphic System (originated by Strombert-Datagraphix Inc. and the Rand Corporation) has been developed. This graphic module is essentially a general purpose picture-editing graphics system which provides the MC/DG user with sophisticated interactive graphics capabilities to display a retrieved calculated/analyzed data set via an on-line graphics terminal. This module is equipped with easy-to-use graphic command syntax sets, and is flexible enough to permit users to plot, interactively, the selected data set using a chosen graphic format.

2. IGS LIBRARY OF SUBROUTINES

Extensive usage of the IGS subroutines is incorporated into this graphic display module to ensure an adequate selection of display formats. The basic geometry library subroutines provided by the IGS system may be used to display joined line segments (LINESG), independent line segments (SEGMTG), plotted points (POINTG), circles and arcs (CIRARC). A subroutine to display multiple line segments (MLTPLG) is also provided, facilitating the generation of cross-hatching, shading, grids, multiple dots, etc. Other subroutines are available to draw linear or nonlinear grids (GRIDG) and provide associated labels (LABELG) and titles (TITLEG). A supporting subroutine (SETUPG) may be called to compute appropriate arguments for GRIDG. Another subroutine (GRAPHG) may be used to produce an entire graph (with grid, labels, titles, and plotted data) from a single call.

In addition to the geometric subroutines discussed above, the IGS system library provides a subroutine (NUMBRG) for displaying numeric data in various formats.

Two subroutines are provided for displaying strings of text. One of these (LEGNDG) begins the display at a specified x,y coordinate and the other (TEXTG) begins at the current position determined by the previous display. Both of these subroutines allow modifications to be made (via the

SETSMG subroutine) in character size, line, and character orientation, character spacing, character font and case, etc. Vector characters may be drawn with any of four line widths; their size, orientation, and skew angle are not restricted by the IGS system.

Table 4 itemizes each IGS general graphic subroutine and briefly describes its function.

3. HARDWARE

(a) Graphics Terminals

The direct-view storage tube (DVST) has been designed to avoid the need of refreshing the cathode ray tube (CRT) graphics output devices from data stored in the graphics memory. Anything displayed on this type of screen will remain visible for up to 1 hour, before it fades. This DVST display device is, in some respects, less versatile than the CRT display device, but also significantly less expensive to purchase and to maintain. In addition, it has higher resolution, less flicker problems, and is more suitable for time sharing remote users than CRT graphics display devices. Based on these reasons, BASIS graphics system chooses the Tektronix 4010, 4012, and/or 4014 DVST display devices as its graphics terminals which can be operated at selectable speeds of 10, 30, or 240 characters per second.

The Tektronix 4010, 4012, and/or 4014 graphics terminals consist of two or three logical parts: (1) the DVST display screen, (2) the keyboard and a positionable cross-hairs cursor, and (3) an optional Tektronix Hardcopy unit.

The 4012 display screen is 8.5 inches by 6.25 inches in size, and consists of 1024 addressable points on the horizontal, or X axis, and 781 addressable points on the vertical, or Y axis. The lower left corner of the screen is the (0, 0) position. This display tube may be operated in a graphics mode (plotting) and/or an alphanumeric mode (printing). All plotting is under the control of user's software. Printing may be in one character size on 4010 model, two sizes on the 4012, or four sizes on the 4014.

The keyboard is a standard ASCII keyboard, with several additions. There are extra keys for erasing the screen and for causing an electrostatic hard copy to be generated.

TABLE 4. IGS SUBROUTINES

Name	Function
	Initialization, Termination, and Control Subroutines
MODESG	Initializes the IGS system. This must be the first IGS subroutine called by a graphics program.
SUBJEG	Sets up the subject space coordinate system.
OBJCTG	Sets up the normalized object space.
SETSMG	Sets a mode array value.
RSETMG	Resets all parameters of the mode array to their default values.
PAGEG	Controls frame advance and form flash.
EXITG	Terminates graphic output. This must be the last IGS subroutine called by a graphics program.
	Graphic Output Subroutines
LEGNDG	Displays (plots) characters.
LINESG	Draws joined line segments.
MLTPLG	Draws multiple line segments.
NUMBRG	Displays (plots) numeric data.
POINTG	Plots symbols.
SEGMTS	Draws line segments.
TEXTG	Types characters.
	Graphing Subroutines
GRIDG	Draws a grid.
LABELG	Labels a grid.
TITLEG	Titles a grid.
SETUPG	Computers optimum grid parameters.
GRAPHG	Draws a complete graph.
	Conversion Subroutines
CONVTG	Converts a character string to numeric form.
FMTSG	Converts numeric data to a character string.

The cross-hairs position is controlled manually by the two thumb-wheels located on the right-hand side of the keyboard. One of them moves the cross-hairs horizontally, and the other one moves it vertically. Once the cross-hairs have been positioned, striking any key on the keyboard will transmit the current location of the cross-hairs intersection to the computer. These cross-hairs are turned on by BASIS graphics software, and turned off automatically when the intersection location is sent to the computer. Once there is a graph displayed on the screen, it is important to take advantage of the cross-hairs positioning capability to divide the whole screen into two areas, namely a working area and a control area. The working area is defined as the area where the actual graph is to be displayed, and the control area is reserved for BASIS graphics software to print out messages and for the user to continue his graphics requests.

The optional Tektronix hard copy unit is used to generate an electrostatic hard copy of what is displayed on the screen. Approximately 5 seconds are needed to generate a hard copy. The process does not affect what is displayed on the screen, and it always ignores the cross-hair lines.

4. SOFTWARE

The BASIS graphics system consists of mainly two levels of software packages. The higher level software package is the graphics command language interpreter which scans through the user's graphics request and performs necessary interpretation. The lower level software package is the actual graphics software which constructs whatever plot is defined by the user. The graphics output media can be a display plot on the DVST screen and/or a paper hard copy. The Tektronix version uses COMPIO routines as the driver routines for the Tektronix terminals. The COMPIO routines are responsible for vector output, erasing the screen, output horizontal character screen, reading cross-hairs location, ringing the bell, causing a pause, and flushing the command buffer. Thus, it is a user-oriented system that allows the user to think in terms of his display, rather than in terms of the hardware features which actually create the graphics output.

5. SYSTEM CONFIGURATION AND INFORMATION FLOW

Figure 36 is a brief overview of both the system configuration and information flow of the computerized MC/DG graphics module. The

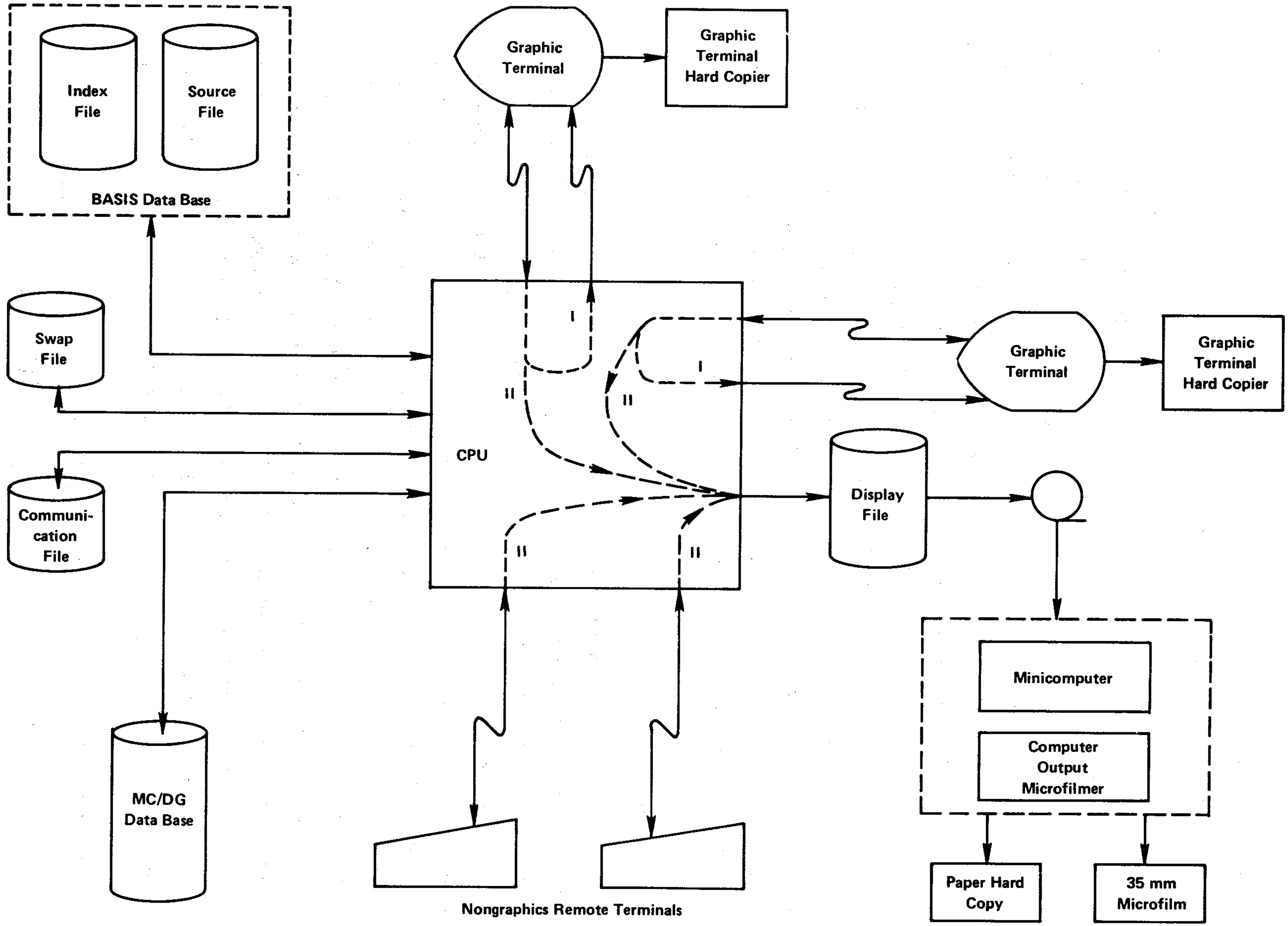


FIGURE 36. SYSTEM CONFIGURATION AND INFORMATION FLOW FOR BASIS PLOT MODULE

Tektronix version always flushes its graphics instructions buffer to the Tektronix DVST via path I.

6. GRAPHICS OUTPUT MEDIA

This user-oriented graphics module allows the user to choose output format. A set of plotted data may be displayed on a graphics terminal (currently using Tektronix DVST).

The set of graphics display commands currently available are:

- PLOTXY - This command permits the user to do a regular XY plot. It sets up the grid scale [(linear-X, linear-Y) or (log-X, log-Y) or (linear-X, log-Y) or (log-X, linear-Y)], draws the graph frame (grids with full lines or tick marks and the axes are automatically labelled in evenly spaced increments), centers and plots the titles (X and/or Y titles and/or plot title) and then performs the actual data drawing signifying desired line specification (solid line or point graph with chosen characters).
- MOREXY - This command permits the user to add subsequent graphs sharing the same graph frame as defined by PLOTXY. It does nothing but draw the actual data specified by the different line specifications.
- BARGR - This command permits the user to do bar graphs. It can be either a vertical or horizontal bar graph. The data which can be used to plot one-dimensional vector (one section per bar with chosen texture).

Detailed description of these command calls, plus a sample output, are given in the following section.

7. MC/DG GRAPHICS COMMAND

An MC/DG graphics command is defined to be any sequence of graphics input elements that activates a process and/or plot within the graphics module. There are two kinds of command elements. The primary command element acts as a verb, i.e., it is always located at the beginning of the command language statement and it defines which kind of process and/or plot is to be performed. On the other hand, the secondary command element acts

as a noun which specifies either the data to be plotted or the plot parameter which overwrites the default value provided by the graphics module.

Table 5 outlines each primary command element with its associated secondary command elements. All the secondary command elements are optional, except those marked with an asterisk. All the required secondary command elements are used to specify the data to be plotted by the user.

The description of each primary command element with its associated secondary command element set includes:

(1) General description of primary command element

(2) General syntax

(3) Description of secondary command elements

(4) Example.

8. PLOTXY COMMAND

a. General Description of Primary Command Element, PLOTXY

This PLOTXY command permits the user to do a regular XY plot. It sets up the grid scale [(linear-X, linear-Y) or (log-X, Log-Y) or (linear-X, log-Y) or (log-X, linear-Y)], draws the graph frame (grids with lines or tick marks and the axes are automatically labelled in evenly spaced increments), centers and plots the titles (X and/or Y titles and/or plot title) and then performs the actual data drawing, signifying desired line specification (solid line or point graph with chosen characters).

b. General Syntax

PLOTXY, X = array name, Y = array name, X1 = array name,
Y1 = array name, X2 = array name, Y2 = array name,
X3 = array name, Y3 = array name, X4 = array name,
Y4 = array name, XTITLE (string), YTITLE (string),
XSCALE = $\begin{bmatrix}\text{LINEAR}\\ \text{LOG}\end{bmatrix}$, YSCALE = $\begin{bmatrix}\text{LINEAR}\\ \text{LOG}\end{bmatrix}$, CHAR (LINE,
CHAR 1, CHAR 2, ...), FRAME = $\begin{bmatrix}\text{TICK*}\\ \text{GRID}\end{bmatrix}$,
XMIN = minimum value of x axis, YMIN = minimum value of
Y axis,

TABLE 5. PRIMARY AND SECONDARY COMMAND ELEMENTS

Primary Element	Secondary Element
PLOTXY	X*
	Y*
	X1
	Y1
	X2
	Y2
	X3
	Y3
	X4
	Y4
	XTITLE
	YTITLE
	PLOTTITLE
	FRAME
	XSCALE
	YSCALE
	CHAR
	LINE
	XMIN
	YMIN
	XMAX
	YMAX
MOREXY	X*
	Y*
	CHAR
	LINE
BARGR	DATA*
	XTITLE
	YTITLE
	PLOTTITLE
	ORIENT
	TEXTURE
	LINE

XMAX = maximum value of x axis, YMAX = maximum value of Y axis,

LINE = BASIS line number, END

Note:

(1) The primary command element (i.e., PLOTXY) is always located at the beginning of the command statement.

(2) Either a comma or a blank serves as a separator between command elements.

(3) All the capitalized words and associated signs (either an equal sign or a pair of parentheses) are the reserved command words.

(4) The user can choose one of those values or reserved words imbedded in a bracket. In case there is no specification, the one marked with an asterisk is the default choice.

(5) X, Y, Y1, Y2, Y3, Y4 and Y, X, X1, X2, X3, X4 are mutually exclusive.

c. Description of PLOTXY Secondary Command Elements

PLOTXY	In case a user wants to make an XY plot, the primary command element PLOTXY is always required to be located at the beginning of the command statement.
X=array name Y=array name X1 Y1 X2 Y2 X3 Y3 X4 Y4	Both are required secondary command elements. The array names for X and Y can be any numerically defined variable names specified in the same BASIS line number. (BASIS line number can be specified by using command element LINE = line number, otherwise the default line number which is always one less than the most current one will be used).
XTITLE(string) YTITLE(string)	A string of characters (up to 60) can be used as the title description for the X or Y axis. All the special characters are acceptable except an

	asterisk which is used as the delimiter of the character string. The default string for XTITLE or YTITLE is the corresponding defined variable name.
PLOTTILE(string)	A string of characters (up to 60) can be used as the title description for the whole plot. All the special characters are acceptable except an asterisk which is used as a delimiter of the character string. The default string for PLOTTITLE is the Data Base name.
XSCALE = $\begin{bmatrix} \text{LINEAR*} \\ \text{LOG} \end{bmatrix}$ YSCALE = $\begin{bmatrix} \text{LINEAR*} \\ \text{LOG} \end{bmatrix}$	Any axis (X or Y) can be in either linear or logarithmic scale. With the combination of these two secondary command elements, the grid scale can be set up as (linear - X, linear - Y) or (log - X, log - Y) or (linear - X, log - Y) or (log - X, linear - Y). The default setup is always (linear - X, linear - Y).
CHAR (LINE, CHAR 1, CHAR 2, CHAR 3, CHAR 4)	CHAR(A) is the default choice which indicates all data points will be plotted by using character A. CHAR (char) indicates all the data points will be marked with the chosen character (any character on the key-board).
FRAME = $\begin{bmatrix} \text{TICK*} \\ \text{GRID} \end{bmatrix}$	FRAME=TICK is the default choice which draws the graph frame in tick marks. FRAME=GRID grids the graph frame with full lines.

XMIN=minimum values for x axis YMIN=minimum values for y axis XMAX=maximum values for x axis YMAX=maximum values for y axis	Since the default values are the minimum and maximum values of the first graph plotted by PLOTXY, user can use any of these secondary command elements to set up the limit values which may be necessary for the subsequent graphs requested by MOREXY.

d. Examples

(1) PLOTXY,X=X,Y=Y, END

(2) PLOTXY, X=X, Y=Y,CHAR(LINE), XTITLE(TEMPERATURE), YTITLE(YIELD STRENGTH), PLOTTITLE(MILS TEST PLOT),FRAME=GRID,END

(3) PLOTXY,X=Y,Y=X,CHAR(X), XSCALE=LOG, YSCALE=LOG, END

(4) PLOTXY, X=X, Y=Y, Y1=Y1, Y2=Y2, Y3=Y3, CHAR(LINE,+,*,1),END

(5) PLOTXY, Y=Y, X1=X1, X2=X2, CHAR(A,B),END

LINE=BASIS line number	For BASIS user, this secondary command element allows the user to specify the desired BASIS line number. The default line number is always one less than the most recent line number.

9. MOREXY COMMAND

a. General Description of Primary Command Element, MOREXY

This MOREXY command permits the user to add subsequent graphs sharing the same graph as defined by PLOTXY. It draws the actual data specified by the same or a different line specification.

b. General Syntax

MOREXY, LINE = BASIS line number, X = array name, Y = array name,

$$\text{CHAR} = \begin{matrix}\text{LINE}\\ \text{Char}\end{matrix} ,$$

NOTE: (1) The primary command element (i.e., MOREXY) always located at the beginning of the command statement.

(2) Please refer to command PLOTXY for other comments.

(3) BASIS line number should be specified right after the primary command element (MOREXY in this case).

c. Description of MOREXY Secondary Command Elements

Please refer to the section which outlines the description of PLOTXY secondary command elements (page 174).

d. Examples

(1) MOREXY, X=X, Y=Y,END

(2) MOREXY, X=X, Y=Y CHAR(A),END

(3) MOREXY, X=YEAR, Y=MP, CHAR(+),END

(4) MOREXY, X=YEAR, Y=MP, CHAR(LINE),END.

10. BARGR COMMAND

a. General Description of Primary Command Element, BARGR

The BARGR command provides the user the options of drawing a vertical or a horizontal bar graph for a one-dimensional array of retrieved computed numeric data.

b. General Syntax

BARGR, X = array name, XTITLE (String), YTITLE (String),

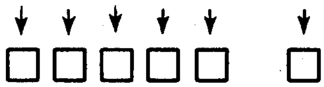
PLOTTITLE (String), ORIENT = $\begin{matrix}V\\H\end{matrix}$, TEXTURE = $\begin{matrix}1\\2\\3\\4\\5\\6\end{matrix}$, END

c. Description of BARGR Secondary Command Elements

BARGR	This primary command must be located at the beginning of the command statement.

Data=array name	The array name can be any numerically defined variable name specified in the same BASIS line number (same as in PLOTXY).
XTITLE(string) YTITLE(string)	A string of up to 60 characters can be used as the title description for the X or Y axis. All the special characters except * are acceptable. The default string for XTITLE and YTITLE are the corresponding X and Y variable names.
PLOTTITLE(string)	A string of up to 60 characters can be used to describe the bar graph. With the exception of the *, all other special characters are acceptable.
ORIENT=V H	For vertical bars For horizontal bars

TEXTURE = 1, 2, 3, 4, 5, or 6



11. TECHNIQUES FOR DISPLAY OF DATA SCATTER AND CONFIDENCE

Let us start by describing the statistical analysis performed by the MC/DG system. The sample MC/DG data base contains a "STATS" subfile whose elements are coefficients of a set of multiple linear regression equations. Multiple regression is the general statistical technique selected to provide the MC/DG users with the capability of inferring or estimating the relationships between a dependent variable and its overall dependence on a set of other independent variables.

Present in the "AVERAGE" subfile are sets of discrete part costs for varying part measurements, material types, manufacturing processes, and designer-controlled complexities. These values are the mathematical averages of the raw data submitted by the five team companies. These data have been carefully reviewed and deemed acceptable before being entered

into the computer. Suppose an airframe designer wishes to estimate the discrete part cost of a specific part, with a different part measurement other than the standard ones. He may invoke the MC/DG data base, enter the appropriate command and its corresponding functions and parameters, an estimated discrete part cost will be displayed to him.

In this application, we are utilizing one of the most important uses of the regression technique as a predictor. The problems of statistical inference can be grouped into two general categories: estimation and hypothesis testing.

a. General Display Techniques

(1) Estimation

The purpose of estimation is to find the more likely population parameters from the examination of sample observation. In our case, we wish to estimate the regression coefficients in the "STATS" subfile from the sample data in the "AVERAGE" subfile and establish some confidence intervals.

(2) Hypothesis Testing

We focused on evaluating various hypothesis about the data, e.g., does manufacturing lot size contribute significantly to the base-part cost variation. One may simply test the null hypothesis that its value is zero against the alternative hypothesis that its value is greater or less than zero.

Some of the most-often used null hypothesis in multiple regression are:

- There is no linear relationship between a dependent variable and a set of independent variables
- A particular independent variable has no linear effect on the dependent variable once the effect of other independent variables are adjusted for
- The relationship between the dependent variable and particular independent variable is nonlinear, and that the effects of two or more variables are not additive.

b. Data Scatter Display

In performing multiple linear regression, values of the dependent variable are predicted from a linear function of the form

$$Y' = B_o + B_1X_1 + B_2X_2 + \ldots + B_iX_i \quad ,$$

where Y' represents the estimated value for Y, B_o is the Y intercept and the B_i's are regression coefficients. (The values of B_i are elements of the subfile STATS). The B_i coefficients are selected in such a way that the sum of squared residuals is minimized, i.e.,

$$\sum_i (Y - Y')^2$$

is minimum. The graph representing the linear function Y' is called the regression line.

Figure 37 shows the general data scatter. It also shows that the constant B_o is the point at which the regression line crosses the Y axis and represents the predicted value Y' when $X_i = 0$. The constant B_i gives the slope of the regression line and indicates the expected change in Y with a change of one unit in X_i. The predicted Y' values fall along the regression line, and the vertical distances (Y - Y') of the points from the line represent residuals (errors in the prediction).

c. Data Confidence Display

It is often essential to evaluate the accuracy of the prediction equation, or equivalently, to determine the amount of prediction error associated with the predictions. This is done by examining one or more of the statistics that reflect the average size of residuals. One may choose to examine the "standard error of estimate" defined by

$$SEE = \sqrt{\frac{\Sigma(Y - Y')^2}{N}}$$

where N = sample size.

This value may be interpreted as an "average error" in predicting Y from the regression equation.

Alternatively, one may establish a confidence interval, i.e., upper and lower limits within which the true value Y lies, e.g., a sample of four observations:

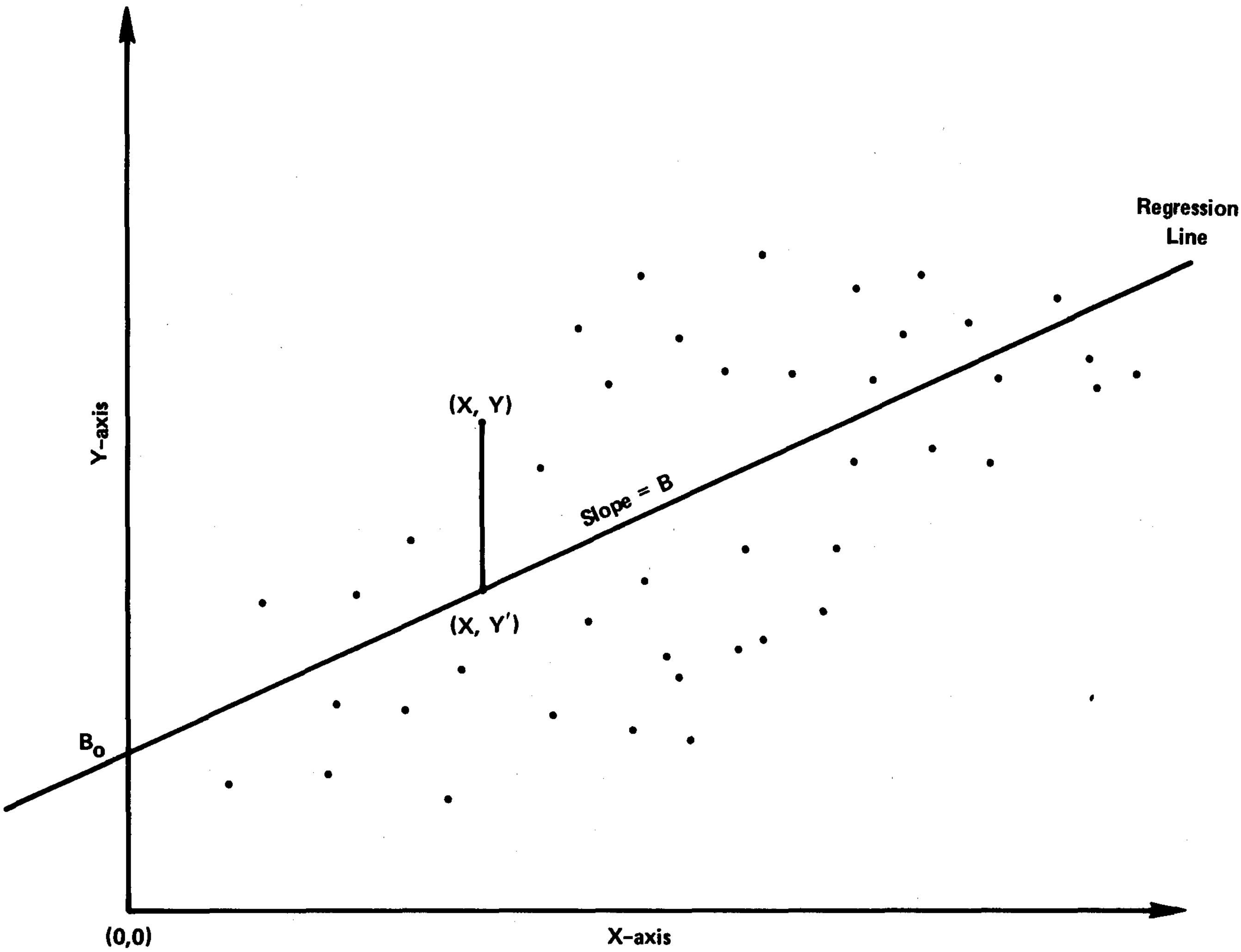


FIGURE 37. DATA SCATTER ABOUT A LINEAR REGRESSION LINE

$$X_1 = 1.2;\ X_2 = 3.4;\ X_3 = 5.6;\ X_4 = 0.6$$

drawn from a normal population with unknown mean μ and known standard deviation σ. The maximum-likelihood estimate of μ is the mean of the sample observations

$$\overline{X} = \sum_{i=1}^{4} x_i/4$$

We wish to determine upper and lower limits which are rather certain to contain the true parameter value μ between them. Since the quantity

$$Y = \frac{\overline{X} - \mu}{\sigma/\sqrt{n}} = \frac{\overline{X} - \mu}{3/2}$$

is normally distributed with mean = 0, standard deviation = 1. Therefore, Y has a density function

$$f(Y) = \frac{1}{\sqrt{2\pi}}\ e - \frac{Y^2}{2}$$

which is independent of the true value of the unknown parameter. To compute the probability that Y will be between any two arbitrary chosen numbers, say 1.96, therefore,

$$P(-1.96 < Y < 1.96) = \int_{-1.96}^{1.96} f(Y)dy = 0.95$$

Now $-1.96 < Y$ is equivalent to $-1.96 < \frac{\overline{X} - \mu}{3/2}$ or $\mu < \overline{X} + 3/2(1.96) = \overline{X} + 2.94$.

Similarly, $Y < 1.96$ is equivalent to $\mu > \overline{X} - 2.96$. Therefore,

$$P(\overline{X} - 2.94 < \mu < \overline{X} + 2.94) = 0.95$$

or

$$P(-0.24 < \mu < 5.64) = 0.95.$$

Thus, two limits have been obtained (-0.24, 5.64), that are 95 percent certain to contain the true parameter value μ between them. This interval, -0.24 and 5.64, is called the 95 percent confidence interval; the probability, in this case, 0.95 is called the confidence coefficient. Similarly, we can obtain intervals with any desired degree of confidence less than 1.

In general, any numbers a and b such that the ordinates f(a), f(b) at those points include 95 percent of the area under the curve of f(Y) will determine a 95 percent confidence interval.

APPENDIX D

USERS GUIDE FOR SAMPLE SYSTEM

APPENDIX D

USERS GUIDE FOR SAMPLE SYSTEM

1. GETTING ON THE SYSTEM

To LOGIN to the computer system, the standard login procedure for the computer system being used follows. The system used at Battelle for the procedure is:

LOGIN, "NAME", "CODE", SUP, X

where "NAME" is the users last name and "CODE" is the user password.

The computer will now ask for a project number and possibly a password, depending on the security arrangements provided for the system. The user types in the project number and password assigned to him.

The user is now connected to the main computer system and must activate BASIS. This is done by the following statement:

/LOGON, BASIS, PHASE II.

The necessary permanent files are attached automatically by preprogrammed systems commands activated by the LOGON to BASIS. The / in front of the characters indicates that the string is a CDC catalog procedure.

2. DATA SEARCHES

The system will respond with information concerning the data base, such as last update and the number of records in the data base, then ask the user for his request. At this point, the user can search using individual requests to the system, or initiate the use of one of the canned procedures from the PROFILE list.

a. Using PROFILES (Saved Procedures)

To use one of the PROFILE procedures, the user would type:

PROFILE EXECUTE "PROFILE NAME"

where "PROFILE NAME" is the name of the PROFILE desired. A list of available PROFILES is included in Table 6. These PROFILES, initially created by the developer of the computerized MC/DG, and modified or expanded by industrial users as they become familiar with the system, would normally contain procedures to search for the desired data and

TABLE 6. LIBRARY OF PROFILES ILLUSTRATED USING PROFILE SHOW COMMAND

```
PROFILE SHOW

  **** AVAILABLE PROFILES ****

  ****** SAVE FILE ******
ID        .NAME.
A1    SELECT
A2    START
A3    ANALYSIS
A4    POWER
A5    ASYMPTOTIC
A6    ANALYSIS PANEL
A7    POWER PANEL
A8    ASYMPTOTIC PANEL
A9    COSTS TRANSFORM
A10   RESIDUALS
A11   CALC COSTS VS LOTSIZE
A12   NEW LOTSIZE
A13   ANALYSIS LOTSIZE
A14   SELECT VS LOTSIZE
A15   PLOT VS LOTSIZE
A16   SELECT VS LENGTH, LOTSIZE
A17   CALC COSTS VS LENGTH, LOTSIZE
A18   ANALYSIS LENGTH, LOTSIZE
A19   SELECT COMPANY
A20   TAB CALC COSTS VS LENGTH, LOTS
A21   RESIUDALS LENGTH, LOTSIZE
A22   POWER VS LENGTH, LOTSIZE
A23   NEW C3
A24   LINEAR VS LENGTH, LOTSIZE
A25   TAB STATS LENGTH, LOTSIZE
A26   CALC L0,L1,L4,L5,C1
A27   LINEAR VS L0, L1
A28   ANALYSIS COMPANY, LENGTH, LOTS
A29   LINEAR VS L0, L1, L4
A30   REDI
A31   STATS L0, L1, L4
A32   ANAL PANEL COMP, LENGHT, LOTSI
A33   CALC L0,L1,L4,L5,L6,C1
A34   LINEAR VS L0,L1,L4,L5,L6
A35   STATS L0,L1,L4,L5,L6
A36   NEW WIDTH
A37   PLOT VS WIDTH
A38   NEW AREA
A39   PLOT VS AREA
A40   GROUP COUNT
A41   AN03R PANEL COMP, LENGTH, LOTS
A42   SELECT NULL
A43   CURVES=1 VS L0,L1,L4,L5,L6
A44   DELETE DV
A45   PLOT VS NULL
A46   NEW NULL
A47   SAMPLE X-Y PLOT
A48   START 1
A49   NEW LINE 1
A50   SELECT DATA
A51   MCDG SAMPLE PLOT VS LENGTH
A52   EXTRAP COST
A53   SAMPLE USAGE
A54   COEF OUTPUT
A55   COMPARISON BAR CHARTS
A56   SELECT DRIVER AND GRAPH
A57   SELECT LINEAL METHOD
A58   SELECT LINEAL MATERIAL
A59   SELECT LINEAL LOT SIZE
A60   SELECT LINEAL LENGTH
A61   IDENTIFY

  ******  PROFIL1  ******
ID        .NAME.
B1    CHOOSE
B2    XXX
B3    CALC
B4    TABLE
B5    COSTS
B6    PLOT
```

then display that data in a usable form such as x-y graphs, text, pie charts, tables, or bar charts. For example, the command

PROFILE EXECUTE COMPARISON BAR CHARTS

would initiate a procedure to search for the data required by the user, specified by asking the user a series of questions, then plotting the data as bar charts. A listing of cataloged PROFILES can be obtained by typing

PROFILE SHOW.

To view the text of a particular profile the command entered is

PROFILE SHOW <NAME>

where <NAME> is spelled and punctuated exactly as when saved. The procedures necessary to create or edit a PROFILE are detailed in the BASIS Users Guide.[1]

b. User-Directed Searching

If the user does not want to use one of the PROFILES, he may search for the information himself. The most efficient method is to use the system index files. An extention of the index term capability provided by BASIS, that will be most useful to the MC/DG user, is called stem selection. By entering part of a search term and following it with an asterisk, BASIS will search the index term file for all terms with that common prefix or stem. For example, the stem request "MATERIALS*", can be used to find all the materials included in the data base. BASIS will respond to a stem selection request by listing, with labels A to Z, up to twenty-six terms with the requested index prefix. The user may select a subset of the listed terms using the following notational options:

- A,B,C,E (selections separated by commas)
- A TO C,E (selection of a range of terms using "TO")
- ALL (selection of all terms listed)
- MORE (request continuation of term listing if list not completed).

(1) BASIS Users Guide, Information Systems Section, Battelle's Columbus Laboratories, Columbus, Ohio (1976).

As indicated by the prompting message "SELECT TERMS, OR ENTER REQUEST", the user may continue making selections, each of which is tagged by a line number, until a new request (command or search term) is entered. All terms selected by a stem selection are automatically ORed together. If the user does not wish to select from the stem list, a command or retrieval request may be entered. You may also limit the number of terms that will be displayed and choose which stem in the series with which to begin the display by entering two numbers immediately following the stem selection, for example:

MATERIAL,*5 would list only 5 stems, even if more existed

MATERIAL,*6,3 would list only 3 stems, beginning with the sixth stem found.

For example, if the user wished to look at what materials were available, the request would be MATERIAL,*. The system would respond with a listing like the one below. The user would then select the material he wants to search by typing the letter corresponding to that material. In this example, "D" (Steel) is the desired material, and the computer responds to tell the user the number of items in this classification in his data subject:

```
 .ITEMS.     TERM
 IN YOUR DATA SUBSET
A      536  MATERIAL:ALUMINUM
B      536  MATERIAL:ALUMINUM-2024
C      244  MATERIAL:PH15
D      244  MATERIAL:STEEL
E      244  MATERIAL:STEEL - PH15 - 7M0
F      368  MATERIAL:TITANIUM
G      368  MATERIAL:TITANIUM - 6AL - 4V
H      536  MATERIAL:2024
I      368  MATERIAL:4V
J      368  MATERIAL:6AL

K      244  MATERIAL:7M0
END OF TERMS WITH YOUR STEM
ENTER LETTERS TO BE COMBINED,
SEPARATED BY COMMAS, OR ALL
   / D
       244 ITEMS
 IN YOUR DATA SUBSET
```

The user may request that all terms with a common prefix be combined by following the stem selection with ALL, thus MATERIAL,*ALL would combine all terms with the common stem and print only the final result. The resulting document set will indicate how may terms were combined in its formation.

Another method of examining the contents of the index is index browsing, in which a stem is followed by two asterisks, again with an optional display limit and start specification. Index browsing operates exactly like stem selection, except that it will not stop when the common prefix is exhausted, but will continue until the required number of index terms has been displayed. If you do not enter a number of terms to display, BASIS will list 26 at a time until the entire index is exhausted or you enter a new command.

c. Hierarchical Searching

It is sometimes useful to limit the universe of a search to the last set of documents selected, so that the universe gets smaller and smaller with each selection; for example, if the user wanted to find the cost information for a straight channel (Part Code 2A) made of steel using the room temperature brake forming method. Since you have no way of knowing how many abstracts (if any) exist on this topic, you may want to begin by selecting the most general subject areas and continually refining the request until the document set has been reduced to an appropriate size.

The drawback to this approach is that if your document set becomes too small at some point, you must back up and revise your approach, and have no real way of knowing which search term it was that restricted your set more than desired. For some applications, however, it can be useful.

To use the HIERARCHICAL search mode simply enter the command:

SET HIER ON

Your next selection will draw from the entire data base (unless you have already selected a universe subset). From then on each document set you select will become the universe for the next selection. When you wish to return to normal search mode, simply enter the command:

SET HIER OFF

Document sets selected in HIERARCHICAL mode carry the annotation "IN YOUR DATA SUBSET" to remind you that you are not searching the entire data base.

The example of finding the cost data for a straight steel channel manufactured by the room temperature brake forming method would proceed as follows.

With the HIERARCHICAL search mode set on, the first request term would be MATERIAL, the second would be PARTCODE, and the third would be MFGMETH (manufacturing method) as shown below.

```
2/ MATERIAL:*

 .ITEMS.    TERM
IN YOUR DATA SUBSET
A      536  MATERIAL:ALUMINUM
B      536  MATERIAL:ALUMINUM-2024
C      244  MATERIAL:PH15
D      244  MATERIAL:STEEL
E      244  MATERIAL:STEEL - PH15 - 7M0
F      368  MATERIAL:TITANIUM
G      368  MATERIAL:TITANIUM - 6AL - 4V
H      536  MATERIAL:2024
I      368  MATERIAL:4V
J      368  MATERIAL:6AL

K      244  MATERIAL:7M0
END OF TERMS WITH YOUR STEM
ENTER LETTERS TO BE COMBINED,
SEPARATED BY COMMAS, OR ALL
  / D
       244 ITEMS
 IN YOUR DATA SUBSET
SELECT TERMS, OR ENTER YOUR REQUEST
  3/ PARTCODE:*

 .ITEMS.    TERM
 IN YOUR DATA SUBSET
A       32  PARTCODE:1A
B       32  PARTCODE:1B
C       32  PARTCODE:2A
D       32  PARTCODE:2B
E       32  PARTCODE:3A
F       32  PARTCODE:3B
G       36  PARTCODE:4
H       16  PARTCODE:5
END OF TERMS WITH YOUR STEM
ENTER LETTERS TO BE COMBINED,
SEPARATED BY COMMAS, OR ALL
  / C
        32 ITEMS
 IN YOUR DATA SUBSET
SELECT TERMS, OR ENTER YOUR REQUEST
  4/ MFGMETH:*

 .ITEMS.    TERM
 IN YOUR DATA SUBSET
A       16  MFGMETH:BRAKE
B       16  MFGMETH:BRAKE FORM-COLD
C       16  MFGMETH:COLD
D       16  MFGMETH:FORM
E       16  MFGMETH:PRESS
F       16  MFGMETH:RUBBER
G       16  MFGMETH:RUBBER PRESS
END OF TERMS WITH YOUR STEM
ENTER LETTERS TO BE COMBINED,
SEPARATED BY COMMAS, OR ALL
  / B
        16 ITEMS
 IN YOUR DATA SUBSET
SELECT TERMS, OR ENTER YOUR REQUEST
```

d. Range Searching

A search term might also be for a numeric range that was not explicitly indexed. In these cases, the index is used in conjunction with a numeric data file to produce the desired results. This form of searching is called range searching and is exactly like a regular index search except:

(1) Only numeric fields can be range searched, and they must have been declared to be range fields when the data base was built.

(2) A considerable variety of syntax is allowed in the search term including the use of relational operators such as GT (greater than), LT (less than), etc., as well as specific ranges. The allowed operators are listed in the following table:

OPERATOR	MEANING	SAMPLE
EQ	Equal to	WT EQ 198 or WT 198
LT	Less than	WT LT 136
LE	Less than or equal	WT LE 135
GT	Greater than	WT GT 119.7
GE	Greater than or equal	WT GE 101
ALL	Select all	WT ALL

Assuming that we have a weight data element in our data base named WT and it was declared as a ranged field when the data base was built, we might use the following search terms.

WT 100/147	would select all documents with weight between 100 and 147 inclusive
WT 100 TO 147	is the same as WT 100/147
WT GT 150	would select all documents with weight greater than 150
WT ALL	would select all documents having a weight data element.

3. DISPLAYING RETRIEVED DATA

a. The BASIS Computational Module (COMP)

The COMP module is used to view selected record contents and to evaluate computational expressions over sets of records retrieved by the BASIS storage and retrieval module. The COMP module is accessed by the DISPLAY, DEFINE, or PRINT commands. The DISPLAY command allows requested expressions to be displayed at the interactive terminal, the PRINT command

causes the results to be printed on the off-line (batch) printer, and the DEFINE command saves subexpressions or results to be used in SAR module (Statistical Analysis Routines) programs or OWNCODE module (user created) programs.

b. COMP Language

The syntax of the COMP language is similar to standard FORTRAN, but knowledge of computer programming is not required. Language expressions are separated by commas. Expressions are comprised of field mnemonics or field numbers, defined variables, and literals (constant values) in conjunction with COMP operators and functions. Field mnemonics or field numbers identify the contents of a record entry (field). Defined variables are expressions that have been evaluated and assigned names (the syntax is: name=expression). The syntax for literals is: "constant". When an expression is evaluated, the BASIS system accesses each record of the retrieved set, addresses for each record the values of all record entries and defined variables, performs the requested operations and functions, and disposes of the results (by displaying, printing, or saving).

The classes of expressions accommodated by the COMP module are:

- Integer (maximum magnitude determined by hardware)
- Floating point (maximum magnitude determined by hardware)
- Logical (true and false values)
- String (full range of character set).

Defined variable names may be assigned to any class of expression.

The evaluation of expressions may be qualified by use of the computational IF statement. Using ε_i to symbolize the ith expression (including defined variable expressions) the syntax of the IF statement is:

IF <logical expression> Then $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_i$ ELSE $\varepsilon_{i+1}, \varepsilon_{i+2}, \ldots, \varepsilon_j$.

Note that the expression immediately following IF must be a logical class expression; all other expressions, ε_i, may be of any class.

c. COMP Operators

The three categories of operators available for use in the COMP language are:

- Boolean
- Relational
- Arithmetic.

The operators are shown in the figure below. The periods on each end of the alphabetic symbol are required for Boolean and relational operators.

Boolean Operators

.NOT.	.AND.	.OR.

Relational Operators

.GT.	(greater than)	.LT.	(less than)	.EQ.	(equal to)
.GE.	(greater than or equal to)	.LE.	(less than or equal to)	.NE.	(not equal to)

Arithmetic Operators

+ (add)	* (multiply)	**(exponentiation)
- (subtract)	/ (divide)	

Arithmetic Functions

LOG	($\log_e$)	MIN	(minimum)	SUM	(sum)
LOG10	($\log_{10}$)	MAX	(maximum)	CUM	(cumulate)
EXP	(base e)	AVG	(average)	DECUM	(decumulate)
ABS	(absolute)	STD	(standard deviation)	LREG	(linear regression)

General Class Function

VAL (value)

The arithmetic operators can be used only in conjunction with numeric (integer and floating point) operands. The relational operators can be used in conjunction with any class of expression, but the present implementation of COMP recognizes only .EQ. or .NE. for comparisons of string or logical expressions. String expressions may utilize only the relational operators (bit manipulation using .AND. and .OR. is not implemented). For the Boolean operators the operands must be logical expressions, that is, they must evaluate to true (1) or false (0). Any

expression utilizing Boolean or relational operators is of logical class. COMP allows expressions which contain both integers and real numbers, but the resultant expression will be of class real. An expression involving only integers and arithmetic operators will be classed as integer. The user should be aware that the integer expression "7/8" results in the integer value "0".

d. The Display Command

The DISPLAY command is the simplest way to have the contents of a document set printed at your terminal. In order to examine the contents of a document set simply enter

DISPLAY

or

DISPLAY <line number>

where <line number> is the number assigned to the document set that you wish to see. Throughout BASIS, if you do not say which <line number> you want to use, it will assume that you want the last document set created.

BASIS will respond by asking which fields you want to see. You must enter a list of field names or numbers separated by commas or the word ALL if you want to see the entire contents of each record. If the document set you wish to see is relatively small BASIS will print it immediately. If it is large BASIS will ask how many documents you want to see first. This is necessary since many users will receive their DISPLAY results at a CRT terminal and must be able to ask for small segments at a time in order to fit each segment onto a screen. Exactly what size document set is "small" or "large" is set by your data base administrator and can be changed.

In response to BASIS' question "HOW MANY DO YOU WANT TO SEE FIRST" (or "HOW MANY" if you are asking that documents after the first group be displayed) you may enter one or two numbers separated by a comma. The first number is the number of documents you want displayed, the second number is optional and tells BASIS where you want to start in the document set. Thus, if you enter

/5

after the prompting slash, BASIS will give you the next 5 documents. If you enter

/<u>5,6</u>

BASIS will give you 5 documents, starting with the sixth document in the set.

A special form of DISPLAY may be used to get a document using its accession number, in which case you need not have that document in a set. If you enter the command

DISPLAY (<accession number>)

BASIS will print the document having the <accession number> enclosed in parentheses. This form of DISPLAY is not used very often since you usually do not know the accession number before the DISPLAY.

The actual format that will be printed by DISPLAY is affected by a number of parameters which may be modified. These parameters and their modification are discussed in conjunction with the SET command (see BASIS Users Guide).

e. The Print Command

Sometimes it is useful to have the contents of a document set printed on an off-line printer rather than at your terminal. The PRINT command allows you to do this. PRINT is used like DISPLAY with two differences: first, your results will not be shown at the terminal, they will be printed on a high speed printer; second, in order to get the off-line print sent to you, BASIS will ask for a mailing address. Simply type in your name and mailing address, separating each line of the address with a colon (:).

As with the DISPLAY command, the format of the output is affected by parameters that may be modified with the SET command (see BASIS Users Guide).

f. The Define Command

The DEFINE command saves subexpressions or results to be used in SAR module (Statistical Analysis Routines) programs or OWNCODE module (user created) programs.

An example of the use of the DEFINE command might be to create a discrete part cost using the base part cost plus selected Designer-Influenced Cost Elements (DICE). For example:

DPCOST = BPCOST + JOGGLES* NJOGGLE + HTREAT

4. USING NON-BASIS PROGRAMS WHILE SAVING RETRIEVED DATA

a. The RUN and XEQ Commands

Occasionally a user will need to perform some task which is not handled by BASIS itself. These tasks may be handled by special software designed to work with BASIS in which case the RUN command is used, or by other software that is installation dependent, in which case XEQ is used. The exact manner in which these commands are used depends on the task you are going to perform and should be documented with the software that you are going to RUN or XEQ. The general form of the commands are

RUN, <task name><parameter list>

XEQ <task name><parameter list>

b. The SUSPEND and ADVANCE Commands

Another method of temporarily leaving BASIS to use the standard computer system and return to BASIS is by using the SUSPEND and ADVANCE commands. The SUSPEND command allows you to temporarily halt your terminal session and have the status of your session saved (document sets, sequential search lines, SET parameter values, etc.). When you enter the command

SUSPEND

your status will be saved and you will be given to the host operating system. When you wish to resume work with BASIS, simply enter the commands

EFL,50000

ADVANCE

and BASIS will begin where it left off.

c. Providing Data to Programs Outside BASIS

The retrieved data from BASIS may be put in a form that can be accessed by user-supplied programs using the Format Report Generator

Module of BASIS. The FORMAT report generator module enables a user to display data elements in a flexible, easily specified manner. Data elements may be user-supplied text, special counters, displayable data from the data base or defined variables. A set of carefully chosen default values simplifies the specification of reports without limiting the user to restrictive fixed formats, and report specifications may be saved, modified, and reused when generated within a profile. Further information on the FORMAT report generator module is available in the BASIS Users Guide.

5. GETTING OFF THE SYSTEM

When you want to stop using BASIS, you may do so using either the QUIT or LOGOUT commands.

QUIT simply ends the BASIS program and returns you to the host operating system. You may then interact with the operating system in any way you like, but must remember to take whatever action is required by that system to end the terminal session.

LOGOUT will end the BASIS program and disconnect you from the host operating system. For users whose only computer interaction is through BASIS, LOGOUT is most commonly used.

APPENDIX E

A DESCRIPTION OF THE DATA BASE MANAGEMENT SYSTEM USED--BASIS

APPENDIX E

A DESCRIPTION OF THE DATA BASE MANAGEMENT SYSTEM USED--BASIS

1. BASIS SYSTEM ARCHITECTURE

BASIS is a modular software system. Figure 38 illustrates the interrelationships between the BASIS EXECUTIVE module and the various application modules. Many of the modules are independent software systems capable of interacting with other BASIS modules directly. A system loader transparent to the user handles transfer of control between the modules. These modules are described in the following sections.

2. DATA BASE STRUCTURE

A BASIS data base may consist of up to seven different kinds of files, each of which performs special functions.

(1) Source data file - contains the actual source data records

(2) Table file - contains the description and location of each of the other data base files

(3) Index file - contains information needed for the retrieval of the source data

(4) Range file - contains numeric range values and other information required for searching by numeric data ranges

(5) Communication file - contains all information for a session

(6) Message/dialogue file - contains all messages for user/system dialogue

(7) Thesaurus file - contains the general thesaurus and the on-line thesaurus for the control of indexing vocabulary.

3. SOURCE DATA FILE

The Source Data File consists of source information records. They are the basic entity of data which passes to and from the application programs under the control of BASIS. Its file format is numeric keyed (NK).

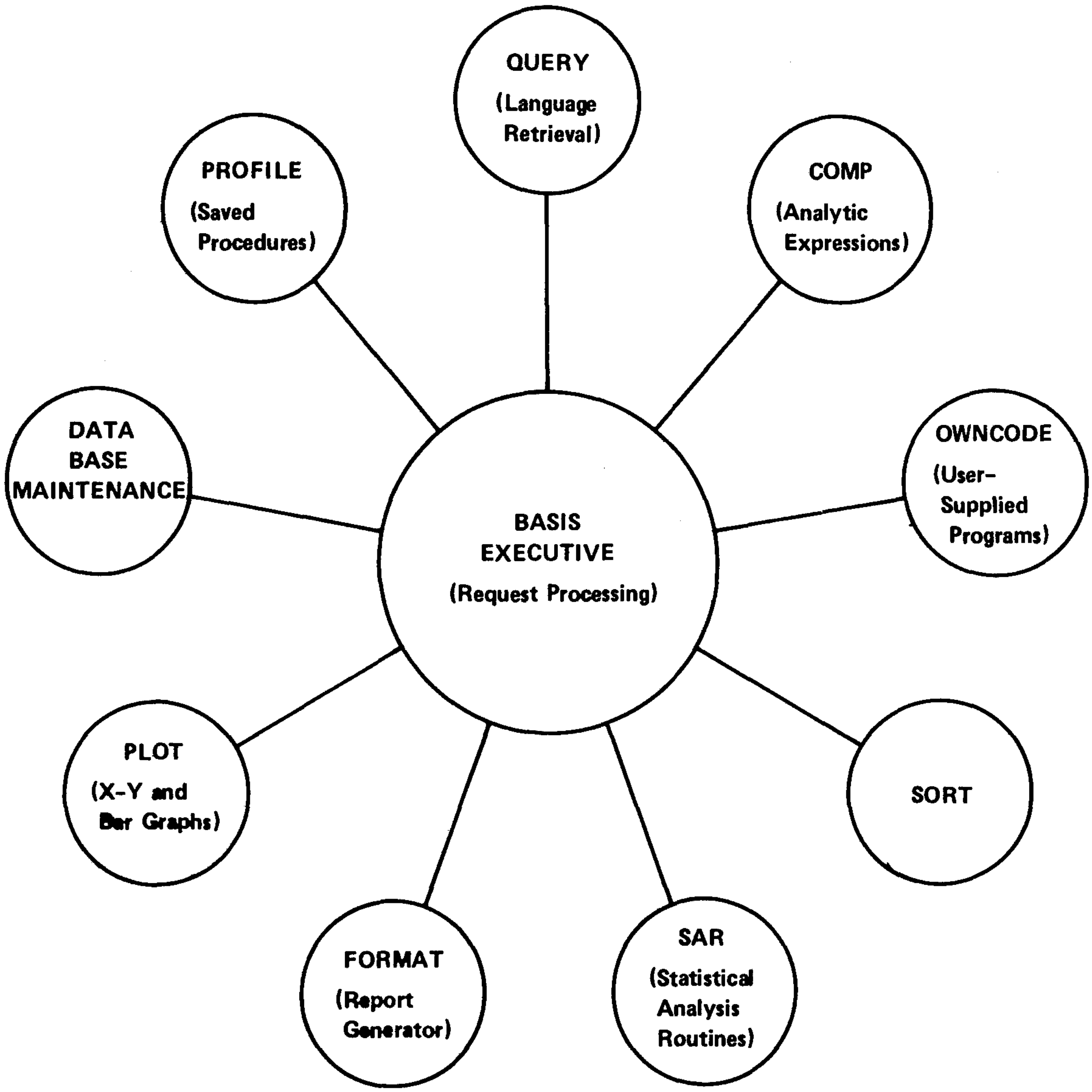
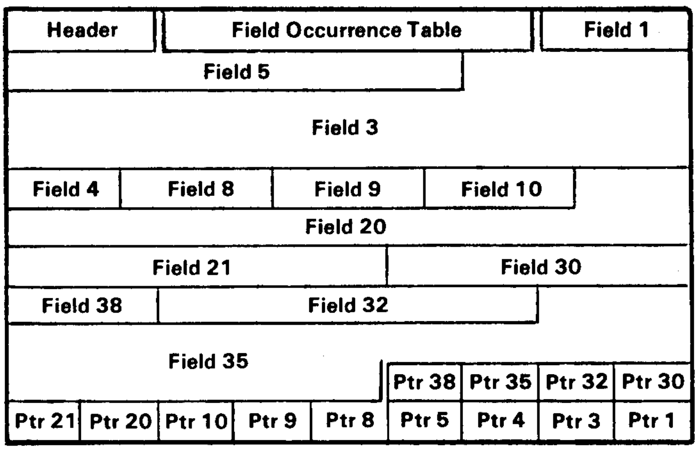


FIGURE 38. BASIS SYSTEM ARCHITECTURE

Each record is accessed by a unique key (called the accession number) that the user specifies at the data base definition time. Each record may contain up to 8000 fields or data elements (the basic units of named data). Each data element is stored as a variable length string of alpha characters (text) even though they may be binary integer numbers, real floating point numbers (decimals), or other binary coded data.

Unlike other DBMS's, BASIS users are not burdened with defining "logical record types". Each record stands alone, or may be logically linked to another record by an application view of the data base.

The format of a sample record in this file is shown below.



BASIS Source Data File Record Format

- **Variable Length Record**
- **Variable Length Fields**
  (stored in bytes, no trailing or leading blanks)
- **Variable Occurrence Fields**
  (occurrence table has a bit to indicate existence of field)
- **Header Word Contains Security Code and Length of Occurrence Table**
  (this allows future definition of new fields)
- **Field Pointers Indicate Where Field is in the Record**
- **Very Compact Record Format**

CELL	DESCRIPTION
HEADER CELL	First word of Source DATA FILE record; Security access code (left most bits) Record Type (bit 16 from right) Highest occurring field number (right 15 bits)

CELL	DESCRIPTION
FIELD OCCURRENCE TABLE	Size, in bits, must be at least as great as the highest occurring data field, rounded upward to full-character multiples. Cell starts in second word of compressed SOURCE DATA FILE record. The pointer flag for data field 1 would be the left most bit, and the pointer flag for data field 30 would be the thirtieth bit from the left in this cell.
FIELD CONTENTS CELL	One cell per field stored, position and length of cell indicated by corresponding pointer cell. The first field (data item) stored (not necessarily the lowest field number) starts to the right of the field occurrence table.
POINTER CELLS	Size is machine dependent--normally 24-36 bits. There will be exactly one pointer cell per field stored (one or two per word) from right to left, starting with the last word of the record. Location of first character of the data field (character 1 is the left most character of the first word of the SOURCE DATA FILE record) (left half) Length of the data field, in characters (right half).

4. INDEX FILE

An index is a guide to the information in a document or in a collection of documents. A searcher in need of information consults the index available to him, and through proper use of it, is guided or directed to the source or has the source pointed out to him or located for him. When a collection exists without an index, retrieval of information must be accomplished by searching the documents themselves.

The filing schemes maintained by most professional persons for their own collections actually are forms of classification indexes. The file folder labels are index descriptions, and indexing for these schemes consists of filing each document in the folder whose index description is considered most appropriate.

Retrieval of information using the BASIS index is accomplished by locating those documents which were identified at the time of indexing by terms representing concepts on which we now want information, e.g., the list of terms by which a document was indexed may be A, B, C, D, AND E. This document should be retrieved when we conduct a search for all sources in which B and D have been topics for consideration.

The Index File is a symbolic keyed (SK) file. Every unique term used in the data base index is a key in the Index File. Each key will locate a group of up to 676 posting records; the postings identify the records in the data base that are associated with the index term. From 1 to 676,000 postings may occur for each term.

A posting references a record in the data base, and it is made up of the record accession number and several optional pieces of data that are in an "index code". These other pieces of data are used to describe some of the characteristics of the record that BASIS needs to know when using the Index File. The index code is divided into the following:

(1) Link Code

(2) Security Code

(3) Subsection Code

Each of these codes may be present without any of the other codes. The entire index code may be absent. The data base designer decides which of these codes should be used.

The link code is used to link index terms. Association links, or simply "links" as they have come to be called, are a means for subdividing a document, which conceivably could have been written as a number of documents (one for each identifiable intellectual relationship), into individual intellectual relationships. If we assume that one document discusses four work measurement techniques, four separate plant visits in a trip report, or four fabrication techniques for four different types of gears, then four documents could have been written instead of one.

The index security code is used to indicate the access restriction that is present for a record referenced by a posting. This code is compared to a user's access code, and the user may or may not be allowed to retrieve the corresponding record or field.

The subsection code is used to divide the data base into sections of interest. Each record in the data base is assigned a code, and the data base is partitioned according to these classifications. A user may search the sections of the data base that are of interest.

5. INDEX SEARCH AND INDEX BROWSING

An index term may be any of the following:

(1) Free-text. A single word found within a data element with the exception of words appearing on a stop word list, i.e., words that do not convey significant textual meaning, e.g., and, or, of, the, etc. The stop word list is chosen by the data base administrator.

(2) Data element. A basic data item within a data record. It may be textual or numeric and of one or more words.

(3) Numeric data range. A predefined data range for a specific data element such as base part setup cost between 3 to 5 man-hours. A record is indexed with a numeric data "range term" if the data element has a numeric value within the predefined range.

(4) Index browsing. A user may enter a stem followed by two asterisks "**"; a list of index terms that follow the stem alphabetically will then be displayed. The user may page through as many lists as he wishes or select and combine the terms from a list.

A "search term" may be a numeric range that was not predefined as an index term. BASIS is able to quickly answer this request via the index and a numeric data file. A search term may be a string of characters entered from the terminal requesting the set of records that contain a specific value. Alternately, a search term may also be a text word that was not entered into the index. In such an instance, BASIS will sequentially search through a subset of records specified by the user.

For example, if the user enters the search term "PARTCODE EQ 1A" which is not a predefined index term, BASIS answers "No such term, want adjacent ones". User may type "YES" then BASIS may print the adjacent terms.

The range value records are variable length records that vary in length from 2 to 502 words. Each record contains the actual data values for numeric data elements that are range searched. A total of 676 range value records may be associated with a range term, and each value in the group corresponds to a posting in the Index File.

A maximum of seven values may be packed into one word of a range value record. This allows 3,507 individual data values to be present in one record. The values may be binary integer values, floating point (real) values, or biased real numbers that are stored as integers.

BASIS carefully controls updates to the Range File and Index File, so that the posting records and the range value records always contain the correct corresponding postings and values. This quality control is taken care of by the Range File Managers.

By storing the range values in one file and their corresponding postings in another file, some useful capabilities and efficiencies are realized. Much less storage is required this way rather than carrying the data values in the posting or the posting in the range value records. No unnecessary redundancy exists in the files. If, for some unusual reason, the Range File was not available, but the Index File was, a user could still retrieve the range terms by using them as they appear in the Index File.

6. RANGE FILE

The Range File is used to support numeric range searching. It is a symbolic keyed (SK) file. The keys are predefined legitimate numeric range terms that have been constructed by the DDL. Keys may be from 10 to 40 characters, and they locate "range value" records. If a user wants to retrieve records based on the numeric value of their data elements, he may declare the limits of the values for those elements that exist in the data base. To identify the data elements that correspond to specified numeric range terms, a "prefix" must be defined in the DDL.

The Range File is always used in parallel with the Index File. Every change to be made for the range terms in the Range File and Index File is described by a range term transaction.

7. NUMERIC RANGE SEARCH

Important features of numeric range searching are:

- The ranges of values are specified by user at search time
- Relational operators: greater than (GT)
  less than (LT)
  equal to (EQ), etc.

  may be used to designate the range
- Considerable variety of syntax is accepted, e.g., LENGTH, 6/8 and LENGTH 6 to 8 will result in the same retrieval.

BASIS assigns ascending line number as reference tags for retrieved record sets. Boolean logic statements may be utilized to combine these sets to perform higher level searches. For example:

- If a user wishes to retrieve all the aluminum, steel, and titanium, angle constant sections (partcode is 1A) that are 144 inches long, manufactured by brake-form method, with base-part man-hours less than one.

```
ENTER YOUR REQUESTS ONE AT A TIME
1/ PARTCODE:1A
     664 Items
2/ LENGTH EQ 144.0
    1300 ITEMS
3/ MFGMETH:BRAKE FORM
     816 ITEMS
4/ (1 and 2 and 3)
      24 ITEMS
5/ BPCOST LT 2.000
    5684 ITEMS
6/ (4 AND 5)
      24 ITEMS
7/ BPCOST LT 1.000
    3560 ITEMS
8/ (4 and 7)
      22 ITEMS
9/ QUIT
```

BASIS searched through entire data base and successfully retrieved 22 data records that satisfy the search criteria this user specified. These records may then be sorted or displayed, or printed in certain report formats.

8. TABLE FILE

The Table File (or data base description file) is created by using the BASIS DDL (Data Definition Language) compiler. This file contains the description and location of each of the other data base files. It also contains descriptive tables that tell BASIS what it needs to know to process the data base. When a data base is being described, the system designer must use the DDL to specify all the characteristics, the structure, and content of the data base to the system. Each data element is defined, and the manner in which it can be retrieved is declared. When a user requests the use of a data base from BASIS, the Table File bearing the name of that data base is searched for. The data base exists only if that Table File can be located.

9. DATA SECURITY, PRIVACY, AND RECOVERY

Data security information for a data base is specified in the DDL program at data definition time. For each user authorized to use the data base, a password and a retrieval code may be defined. At least one user ID must be specified for a data base. For each user ID defined, a password may be provided.

Each user in a secured data base is assigned an integer access code. There are up to three levels of security in a data base: (1) index level, (2) record level, and (3) data element level. Access at each level may be protected by a security code stored with the data, which is compared to the user's access code. The user's access code determines what portion of the data base he may access. If no access code is specified, zero will be assumed, and he may access the entire data base. Conversely, the entire data base may be password protected.

10. DATA INTEGRITY

The BASIS File Maintenance System (FMS) incorporates a set of utility programs or modules known as file managers. This system provides the user a set of software modules that maintain various files used in creating, updating, searching, and retrieving information that constitutes the data base. The file managers always print statistics summarizing transactions processed during execution of file update. At the user's

request, the FMS will also provide an accurate, thorough audit trail with detailed lists of each modification to the data base. Trace-back methods enable the user to regenerate the original data base in its entirety.

11. SPECIAL FEATURES

a. COMP

The BASIS Computational module offers a wide scope of analytical expression evaluations for sets of retrieved data records. COMP has a simple language that is similar to FORTRAN, including the standard Boolean (AND, OR, NOT), relational (=, ≠, >, <, ≥, ≤), arithmetic (+, −, x, ÷) operators and mathematical functions:

- LOG (log, base e)
- LOG10 (log, base 10)
- EXP (base e)
- ABS (absolute)
- MIN (minimum)
- MAX (maximum)
- AVG (average)
- STD (standard deviation)
- SUM (sum)
- CUM (cumulation)
- DECUM (decumulation)
- LREG (linear regression).

b. SAR

The Statistical Analysis Routines module is a stand-alone package. It can be called directly from the BASIS central interactive subsystem via the RUN command. The regression analysis routines included in SAR are:

- STAT21 - A multivariate, linear regression routine for which subsets of the independent variables may be chosen by user option
- BMD02R - A stepwise, multivariate, linear regression routine in which dependent variables are successively added under F-test control

- <u>BMDO3R</u> - A multivariate, linear regression routine in which the data sample set may be split into subsets and the subsets combined by user options
- <u>CURVES</u> - A regression routine for linear and nonlinear functional forms
- <u>TABULATION</u> - A simple report generation capability that lists defined variables (data defined by user specified analytic operations on data base fields) in columns using a default format appropriate to the defined variable class.

c. EDIT

Retrieved data to be used in numerical analysis often requires editing. The editing functions available are:

- CREATE new define variables of any class (integers, decimals, logical, strings)
- MODIFY the value of an existing defined variable
- DELETE an existing defined variable
- SHOW (list) the values of defined variables
- ITEMIZE the names, class, line numbers, and occurrences of nulls for defined variables.

d. SORT

The SORT module enables the user to sort his retrieved data sets in user-specified order. SORT has the following characteristics:

- The sort keys must be record entry (field) mnemonics or numbers
- Up to nine sort keys comprising 270 characters may be utilized
- The starting character and length (in characters) of the sort key within a record entry value may be optionally specified.

e. GRAPHICS

The GRAPHICS capabilities of the demonstration computerized MC/DG are provided by an interface between BASIS and the Integrated

Graphic System (IGS) originated by Strombert-Datagraphix, Inc., and the Rand Corporation. It provides three types of display:

- PLOTXY - to display a two-dimensional (X-Y) plot. It sets up the grid scale, draws the graph frame, centers and writes the titles, and draws the requested curves utilizing user choices of characters and lines.
- MOREXY - to add subsequent graphics sharing the same graph frame as defined by PLOTXY.
- BARGR - to draw a horizontal or vertical bar graph. The data to be plotted can be either a one-dimensional vector (one section per bar with chosen texture) or a two-dimensional array (many sections per bar with different section textures). A more detailed description of this module is included in Appendix C.

f. FORMAT

The BASIS format report generator module enables a user to display data elements in a flexible, easily specified manner. Data elements to be displayed may be user-supplied text, retrieved data elements from the data base, special counters, or defined variables. A set of carefully chosen default values simplifies the specification of reports without limiting the user to restrictive fixed formats, and report specifications may be saved, modified, and reused when generated within a profile. An output specification combines data element names, formats, and editing symbols to produce almost any desired output.

g. OWNCODE

A great deal of application flexibility is achieved by a user-supplied special-purpose program to perform a unique task that is required for an application. This program is usually written in FORTRAN and can access a data base by using BASIS library subroutines. The OWNCODE program is executed by using the RUN command.

(1) System Transportability

Careful design attention has been paid to insure both portability and modifiability of all system software. Battelle is committed to the

continuing support and development of the system features. Users can be assured that BASIS will remain a state-of-the-art retrieval/analysis system and that its continued evolution will not render existing applications obsolete. Users who have chosen to install this system on their own computers do so with the knowledge that a change of computers, even between vendors, will not require the complete redesign of applications using it (as is often required if vendor or other nonmachine independent software had been used).

In addition to Battelle's installation, BASIS is operational at many organizations within the U.S., and several other countries on five different vendor computer systems:

- CDC 6000 series
- IBM 360/370 series
- DEC 10/20 series
- SIGMA 9
- UNIVAC 1108.

APPENDIX F

GLOSSARY

APPENDIX F

GLOSSARY

(1a) COMP

The BASIS computational module offers analytical expression evaluation for sets of retrieved data records. COMP has a simple language that is similar to FORTRAN, including the standard Boolean (AND, OR, NOT), relational (=, ≠, >, <, ≤, ≥), arithmetic (+, -, x, ÷) operators and mathematical functions.

LOG (log, base e)
LOG10 (log base 10)
EXP (base e)
ABS (absolute)
MIN (minimum)
MAX (maximum)
AVG (average)
STD (standard deviation)
SUM (sum)
CUM (cumulate)
DECUM (decumulate)
LREG (linear regression).

(1b) DATA DEFINITION LANGUAGE (DDL)

The BASIS Data Definition Language (DDL) provides a standard, easily understood method for specifying the structure of a data base and for tailoring the many application specific options in BASIS to the particular data base. The various options allow full use of BASIS flexibility, while defaults minimize the number of options which the user must specify for standard applications. The DDL compiler uses information provided in the source language description to create a Table File for the data base. This Table File provides information about the data base necessary for subsequent creation, update, or access. The compiler may also be used to modify the Dialog File which supplies text for the messages issued by BASIS. The DDL source language listing provides a self-documenting record of the structure, options, and input format for a data base, and a comprehensive set of error diagnostics.

(1c) EXEC

EXEC is the executive control module of the BASIS central system.

(1d) EXPLAIN

The EXPLAIN command can be used while a user is in the central system of BASIS to explain error messages and various other topics.

(1e) FILE ACCESS

The FILE ACCESS module contains information necessary to access BASIS files when information is requested by the user. This module is transparent to the user.

(1f) FORMAT

The Basis format report generator module enables a user to display data elements in a flexible, easily specified manner. Data elements to be displayed may be user-supplied text, retrieved data elements from the data base, special counters, or defined variables. A set of default values simplifies the specification of reports without limiting the user to restrictive fixed formats, and report specifications may be saved, modified, and reused when generated within a profile. An output specification combines data element names, formats, and editing symbols to produce almost any desired output.

(1g) FORMS

The FORMS module provides batch mode update capabilities for a BASIS data base. Both new data and updates are handled by FORMS. Data may optionally be checked for correctness by using the validation feature of FORMS. Duplication control of new data entering the data base is performed by the duplication feature of FORMS. With FORMS, a user can load a data base without writing a single application program.

(1h) INTEGRATED GRAPHIC SYSTEM (GS)
(Originated by Strombert-Datagraphix
Inc. and the Rand Corporation)

This graphic module is essentially a general purpose picture-editing graphics system which provides the MC/DG user with sophisticated interactive graphics capabilities to display a retrieved calculated/analyzed data set via an on-line graphics terminal. This module is equipped with easy-to-use graphic command syntax sets, and is flexible enough to permit users to plot, interactively, the selected data set using a chosen graphic format.

(1i) MAINTENANCE

Maintenance of the BASIS data base files is performed by a set of programs or modules known as the FILE MANAGERS. These are the HEAD FILE MANAGER, the INDEX FILE MANAGER, the RANGE FILE MANAGER, and the THESAURUS FILE MANAGER. They simplify translating information from its original form into a data base usable by BASIS

(1j) MONITOR

The MONITOR module provides detailed information on the usage of BASIS. Features of the MONITOR module are listed below:

- Supplies information about capabilities of BASIS used by an individual

- All MONITOR information includes the time of day (used to analyze user experience, think time, etc.)

- User input is captured

- System responses and actions are captured

- Basic accounting information is captured

- Allows a data base administrator to tell how a file is being utilized

- Provides for detailed analysis of user-system interaction.

(1k) PLOT

The PLOT module interfaces the user to the IGS plotting system and to plotting capabilities within the SAR module.

(1l) PROFILE (SAVED PROCEDURE)

Because of the recurring nature of many functions performed within some BASIS modules, the system provides a capability called PROFILE that allows a user to save search statements, display requests, computational expressions, logic statements, and all other repetitive user text. The user may delete or modify profiles in batch or interactive mode using the PROFILE editor. The saved profile can contain variable expressions to be entered by the user at search time. <u>Any</u> BASIS dialog can be saved in a profile, and profiles can be linked to permit one to execute others (profile nesting).

(1m) RUN

To facilitate the highly modular design of BASIS, a special loading program swaps systems modules. The RUN command is used to swap independent BASIS modules into computer memory.

(1n) SAR

The BASIS Statistical Analysis Routines module is a stand-alone package. It can be called directly from the BASIS central interactive subsystem via the RUN command. The regression analysis routines included in SAR are:

STAT21--A multivariate, linear regression routine for which subsets of the independent variables may be chosen by user option.

BMDO2R--A stepwise, multivariate, linear regression routine in which dependent variables are successively added under F-test control.

BMDO3R--A multivariate, linear regression routine in which the data sample set may be split into subsets and the subsets combined by user options.

(1n) SAR (Continued)

CURVES--A regression routine for linear and nonlinear functional forms.

TABULATION--A simple report generation capability that lists defined variables (data defined by user specified analytic operations on data base fields) in columns using a default column width of 10 characters and a default format appropriate to the defined variable class.

EDIT--Retrieved data to be used in numerical analysis often requires editing. The editing functions available are:

(1) CREATE new defined variables of any class (integers, decimals, logical, strings)

(2) MODIFY the value of an existing defined variable

(3) DELETE an existing defined variable

(4) SHOW (list) the values of defined variables

(5) ITEMIZE the names, class, line numbers, and occurrences of nulls for defined variables.

SORT--The SORT module enables the user to sort his retrieved data sets in user-specified order. SORT has the following characteristics:

(1) The sort keys must be record entry (field) mnemonics or numbers.

(2) Up to nine sort keys comprising 270 characters may be utilized.

(3) The starting character and length (in characters) of the sort key within a record entry value may be optionally specified.

(1o) TEACH

TEACH provides for an interactive dialog with the user which explains (with appropriate examples) the various features available. When the user enters TEACH, the system will list available features and ask the user to select one for detailed explanation. The description provided includes instructions and examples to clarify the BASIS capability. The dialog may be addressed to specific data base applications since the TEACH command is <u>data base dependent</u>.

(1p) XEQ

The XEQ command is used to call other software from the system library or user supplied OWNCODE.

(1q) DATA STRUCTURE

The computer processes sets of information according to certain programmed instructions. These sets are not simply masses of data; they involve important relationships between the data elements. The sets of information are organized into data structures. In its simplest form, a data structure might be a linear list of elements. In more complicated situations, a data structure may be a data base with a great many interconnections. A sophisticated computer system will use a number of different data structures. Requirements concerning how the data is to be referenced and manipulated will dictate how a system should organize information into some structure.

For the MC/DG data, a special input formatting program was written to "transform" data recorded on the MC/DG data collection sheets into appropriate alphabetic character strings to be processed by another program, the "MC/DG Input Processor". This program creates/updates the source data file records/fields.

(1r) DATA REPRESENTATION

To understand the nature of various data structures, one should be familiar with the ways in which data are represented in a computer. Virtually all digital electronic computers have bi-state components. That is, every component can be either of two states: a transistor can be conducting (on) or nonconducting (off); a magnetic core can be magnetized in a clockwise or counter-clockwise direction, etc. In each case, the digit 0 or 1 is associated with off or on state, respectively; hence, it is called a binary digit or bit. A larger amount of information may be represented by using several bits grouped together (called a byte). For example, a pair of binary digits may assume any of four possible values, namely:

00	0
01	1
10	2
11	3

In general, n bits (which may be represented by n binary computer components) may have any of 2^n different values. Different computers may use different number systems, e.g., the IBM 360 and 370 series utilize a hexadecimal number system, meaning each group of four binary bits represents one digit; whereas the CDC computers group three bits to form an octal digit.

The following table illustrates the "binary-coded decimal" (BCD) representation of the hexadecimal and octal number systems within the aforementioned computers.

Octal	Decimal		Hexadecimal
000	0		0000
001	1		0001
010	2		0010
011	3		0011
100	4		0100
101	5		0101
110	6		0110
111	7		0111
001 000	8		1000
001 001	9		1001
001 010	10	A	1010
001 011	11	B	1011
001 100	12	C	1100
001 101	13	D	1101
001 110	14	E	1110
001 111	15	F	1111

This is just one way of representing numeric data. What about textural types of data representation? Again, this differs from one computer to another. In the IBM 360 and 370 series, for example, the hexadecimal numbers 40 and C1, or the binary bit strings 0100 0000 and 1100 0001, represent a blank and the alpha character A. Whereas in the CDC computers, the same information is represented by the octal numbers 55 and 01, or the binary bit strings 101 101 and 000 001.

One of the most important features of a computer is its memory. This is the mechanism which allows information to be stored and retrieved as needed. Depending on how data is to be represented, the memory can be viewed as different units of information. Brief definitions of the standard units are given below.

(1s) DATA ELEMENT

The unit of information which a programmer is concerned with in most cases is the data element. A data element is the smallest unit of named data. It may consist of any number of bytes (characters). A data element is often referred to as a field, data item, or elementary item. In BASIS, each data element can be referenced by a unique "field name" (or "field mnemonic"), or by its "field number". During retrieval, a group of data elements may be referred to as a whole by a given name or number. These collections of data elements are referred to as a mapped field. Any sequence of field numbers may be defined as a mapped field.

In the MC/DG data base, each data element is a string of characters. All of the data values are stored as text. Every data element is viewed as a variable length string. A data element often contains multiple occurrences of data values. Each occurrence is separated by a special character, usually a semicolon(;).

Data elements, in general, are not always strings. They may be binary integer numbers, real floating point numbers, or other binary-coded data. However, for almost all BASIS applications, all of the data elements are

stored as character strings. This eliminates the need to know a number of different characteristics telling how data in the data base is stored. All data to the programmer is simply a packed string, that is, without interspersed blanks.

(1t) RECORD

When a data base is designed, the system designer determines the various data elements that are to be stored in the "data base". These elements are grouped into named collections called records. There are actually several different types and formats of records in a data base. However, a programmer is normally only concerned with the "information record". These are the ones containing values for the defined data elements. The other records in a BASIS data base contain information which allows the system to quickly retrieve and manipulate the data elements. These records are not used directly by an applications program; therefore, the content and format of the record will not be discussed here.

Each individual information record has values for some number of data elements. There is some common characteristic that causes us to store the values together. In the MC/DG data file, the data elements may be part code, material type, base part set-up cost, manufacturing method, etc. For each discrete part, we would have a record that contains values for each of these data elements. More complicated data bases may have some records containing, for example, summary information about a colony of rats that are being observed in laboratory experiments. Related to each of these colony records, we may have several hundred records with each containing data about a single experimental observation.

The information record is the basic quantum of data which passes to and from the application programs under control of BASIS. Each record is identified by a unique "accession number" or key. Any data element defined for the data base may be contained in any information record. The data base designer does not have to define the relationships occurring between data elements. Unlike other data base management systems, one is not burdened with defining "logical record types" in BASIS. Each record stands alone, or may be logically linked to any other record by an applications view of the data base.

All of the information records are variable length. Each data element is also allowed to be variable length, and may or may not occur. A great deal of compaction is present in these records. Only one bit is used to signify whether or not a data element occurs in the record. The actual data is stored at the byte level, with all leading and trailing blanks removed.

An application program will retrieve information records by accession number, and then retrieve individual data elements (fields) as required by field number. Every record and data element may be retrieved in any random order. No knowledge of any data structure or record layout is required. A programmer only needs to know the name/number of the desired data element.

(1u) KEYS

In order to randomly access individual records, we need to be able to uniquely identify each record. A key is a unique piece of information that is used to identify or locate a record on file. Keys may be a number of different types of data. The most common types of keys are binary integer numbers and character strings.

Numeric keys are binary integers. When records can be easily identified using a unique number for each, it is good to use a binary integer value as the key. This type of key will usually require one computer word of storage per key. We define a minimum and maximum numeric key for a file. The BASIS Source Data File uses a numeric key for each information record that is stored. This numeric key uniquely identifies each document in the data base and is called an accession number. In the MC/DG data base, the accession number is a 9-digit number that contains information about the discrete part, e.g., the part code, material type, manufacturing method, part description, its measurements, and manufacturing lot size.

(1v) FILE

A named collection of a given type of record is called a file. In a BASIS data base, there are several different files. The file which an application is most concerned with is the one that contains the information records. This file is called the HEAD FILE or SOURCE DATA FILE. The other files are discussed in Section III, and a detailed description of file structures, in general, is found in this section.

The records, in the SOURCE DATA FILE, contain the most significant data for each entry in the data base. The SOURCE DATA FILE is simply a numeric keyed file, the "key" to each record is the unique accession number of the record. One is able to request the BASIS software to return any information record directly from the SOURCE DATA FILE by simply knowing the accession number. The file can contain any number of records, and each record may be fixed or variable length.

(1w) DATA BASE

A data base is essentially a collection of interrelated information (data) or files stored on some storage media, e.g., disks, drums, or tapes, that is operated by a set of computer programs for varying purposes. Its main function is to provide centralized control of the stored data. The advantages accrued from having centralized control of data are summarized as follows:

- The amount of redundancy in the stored data can be minimized.
- Problems of inconsistency in the stored data can be avoided.
- The stored data can be shared.
- Standards for access and maintenance of the data can be enforced.

- Security of data retrieval can be applied.

- Conflicting requirements of different users can be balanced.

- Provision of data independence, that is, the data is organized so that it is independent of programs; each application is serviced in an optimal manner.

The ideal data base would have absolutely no redundant data, but given the current hardware constraints and user requirements, some useful and intended redundancy will exist in a data base. This redundancy is present to give improved access times, capability to recover from accidental loss of data, and sophisticated cross indexing of information. This is sometimes termed "controlled redundancy". The critical issue here is that the redundancy is intentional. All modifications to the data base are made in a common, controlled, and consistent manner. When a data value is updated, all of the information related to that value is also updated.

Almost every data base will change and grow. A good data base system must accommodate this. A systems analyst typically thinks that the data structure he designs will handle all future requirements. Some space is reserved for future changes. The logical view of the data becomes tied to the way it is stored. This method has failed too many times, causing undue grief. A true data base structure allows one to change storage requirements for data elements, and to add new elements easily.

Programs can be relatively independent of the contents of the data base. As new elements are added, some will usually be able to update the new data. Any programs that need to use the element must be modified. If the storage requirement for an element is extended, rather than truncating the data, some programs may be adjusted to process the complete data element. Easy restructuring of the data base must be provided for as new data elements or new applications using an existing data base are added. The restructuring should be possible without having to rewrite the application programs that already exist. In general, adding new data elements or kinds of records should cause as little upheaval as possible. The ease with which a data base can be changed will have a major effect on the cost required to develop new applications.

Different programs will often have different "views" of the data base. That is, any number of relationships may exist between different data elements and different kinds of information records. Since it is desirable to allow for a number of different views, it is very useful not to force one single inter-data element of inter-record logical relationship on a data base.

One reason for using data base techniques is to permit users to employ information in a way which cannot be precisely anticipated by the application designers. For this reason, information should be organized so that it can be addressed in a variety of ways and can be used to answer a diversity of queries.

Today, a great number of applications are being developed that require the data base structure to accommodate access to the stored information from

interactive terminals. Extremely fast access to this information is often very high priority. The users of this information, in many cases, are not experienced in the use of computers; they want to be able to query their data bases in a natural, easily learned, and easily understood way. With a good system, the cost of making interactive retrievals will usually be much less than making the similar retrievals via some manual means. The MC/DG data base, created by means of BASIS, is aimed at achieving these goals.

(1x) FILE STRUCTURES

The information that is kept in a data base must be organized in a manner that promotes rapid access, compact storage, and easy, efficient modification. The file structures used in BASIS give the organization of information in a data base these attributes. The method used by BASIS provides a means for mapping data elements and record into efficient file structures. The application programmer is free from having to understand the details of how the data is actually stored. In the following paragraph, we will explain the file structures used by BASIS. Please realize that it is not critical for one to understand the file structures we use. One can create data bases and do all kinds of wonderful things with them, without knowing anything about what is explained in this section. This information is presented because many people like to know, in detail, what BASIS is doing with their data. Understanding the file structures often gives one insights into how data can be most effectively used for a particular application.

(1y) FILE CONCEPTS

A file is a logically-connected set of information. It is the largest collection of information that can be addressed by a file name. The file name is a unique name by which a file is identified to the host operating system, to BASIS, and to other application programs. The file name is referred to as a LFN (logical file name) or DDNAME (data definition name). If we want to use a file more than one time, then the file must be a permanent file (i.e., we have told the host operating system to keep our file for future use). For permanent files, the file name is equated to a permanent file name (PFN) or data set name (DSN).

(1z) LOGICAL STRUCTURE AND PHYSICAL STRUCTURE

A logical structure (logical data organization) is mapped on a physical structure (physical storage organization). Files are essentially groups of records. The logical structure of a file is the manner in which the records on the file are related to each other. We use the term file organization to indicate the logical structure of a file. The physical structure of a file is dependent on the storage devise on which it resides. The file is divided into relative physical record units (PRU's), all of equal size. The PRU is the smallest unit of information that is transferred between a storage device and main memory. Clusters of the PRU's are often scattered on the device when the file is stored on a disk, drum, or other randomly addressable storage device. On magnetic tape, PRU's follow each other sequentially from the beginning of information (BOI) to the end

of information (EOI). A unit of information, called a block, is used to map the logical structure onto the physical structure. Blocks are interwoven with both the physical and logical structures of the file. A block may be a partial PRU, or one or more PRU's. Actually, a file may contain several types of blocks. Usually a data block contains one or more records. Physically, a file is divided into PRU's; logically a file is divided into records; data blocks contain records, and are recorded on one or more PRU's. They are used to map the logical data structure onto a physical data structure.

(1aa) RECORD TYPES AND RECORD FORMATS

We recall that a record is a set of related data. It is the unit of information that can be transferred from a file to a program's work area within the computer memory by a file-manipulation routine. BASIS uses several kinds of records. Terms such as "record type" and "record format" are used to define different kinds of records. The length, content, and structure of a record depends on the record type and the record format. The term "record type" is used to define the general structure of a record. It describes the characteristic of a record, but not its content. BASIS uses some of the common record types that are supported by the host operating system of the given computer. However, it also uses the record types required for efficient operation, even if the host operating system does not directly support them.

A fixed length record is the simplest type of record. A fixed length record consists of a specific number of bytes. Each record will require exactly the same number of bytes. If each record is stored, one directly after the other on a file, then any relative record could be located by calculating where it is, since each record is the same size. If every record was 80 bytes and the first record started at byte number 1, then the second would start at byte 81, and the fiftieth record would start at byte 3921. The problem with fixed length records is that each record must be the same size as the longest record in the file. This will often cause wasted space.

Variable length records may conserve total file space since each record is only as long as required. Each record does not have to be the same size as the longest record in the file. Each of the records will have a length in characters (or bytes) associated with it. This is called the record length (RL). When records are placed on a file, the minimum record length is usually defined in characters (MNRL) and the maximum record length in characters (MXRL). For some variable length records, the record length is carried within the record, or is maintained in the file structure that stores the record. For other variable length records, a record terminator is used to delimit the records. The inconvenience with variable length records is that it is more difficult to locate individual records at random. The beginning of each record cannot be calculated like it could with fixed length records. Therefore, variable length records must be stored in more sophisticated file structures if they are to be accessed in a random order.

One special type of record is called a card image. For each host system, these records are stored in a different way. Sometimes the records are

80-byte fixed-length records. More often, card images are stored as variable length records, using one or two terminator bytes. Quite often the terminator bytes are a carriage return (CR) and a line feed (LF). This is normally the case for ASCII (DEC IO) and EBCDIC (IBM 360/370 and SIGMA 9) character sets. For CDC, card images are terminated by filling the last word with zero bytes (at least two zero bytes must occur and up to 11 zero bytes may occur). This type of record on CDC is also termed a Z-type record.

Record format is the term used to specify the content of a record. It is a more specific term than record type. If we know a record format, then the record type is implied. The record format indicates how the information in a record is organized. The method of extracting information from the record can be determined by knowing the format of a record.

BASIS uses card image and variable length record types. Several different record formats are also used.

(1bb) FILE ORGANIZATIONS

It is appropriate to store data that has certain processing requirements in file structures that provide capabilities which complement the methods to be used in manipulating the data. Every file structure has certain advantages and shortcomings; it is important to understand what these are. File organization (FO) is the term used to describe the manner in which the file is logically structured. The file organization designed for BASIS is meant to match the processing requirements of the system to the capabilities of the structures used.

The files that make up a BASIS data base have critical interrelationships that allow efficient and comprehensive retrieval, manipulation, analysis, and maintenance of the data. A description of the BASIS data base structure is presented below.

BASIS uses four different file organizations. These are:

(1) Sequential (FO=SO)

(2) Random (FO=RO)

(3) Numeric Keyed (FO=NK)

(4) Symbolic Keyed (FO=SK)

These organizations are described in the following subsections. How the different kinds of files are used is described below.

(1cc) SEQUENTIAL FILES

In a sequential file, records are placed in the order of presentation to the sequential file manipulation routines. Each record follows the previous record. Given the location of one record, the location of the next record is determined by knowing the length of the given record or by scanning for a record terminator. A sequential file may only be accessed by reading each

record in the same order they were placed on the file. A sequential file can reside either on a magnetic tape or on a mass storage device (such as a disk or drum).

A sequential file is the most compact way of storing data in a file. No space is required for directories or index blocks. All of the blocks on the file are data blocks. Files that are to be sent (spooled) to a line printer, or come from a card reader, are common examples of sequential file usage. A sequential file is used when we know future processing of the records will require us to access each record in the file in the same order it was placed on the file. Individual records cannot be accessed without reading all the previous records. Card-image-type records are almost always kept on sequential files.

A terminal is considered by BASIS to be like other devices capable of supporting sequential files. Each terminal read is like a sequential record access, and each terminal write is like outputting a sequential record. The difference is that we cannot reuse the input or output.

(1dd) RANDOM FILES

Random files (or direct access files) allow us to access the data on a file in any random order. Each block on a random (RN) file has a distinct location and a unique address, making it possible to locate any record without extensive searching. Random files must reside on a mass storage device (disk or drum) that is block addressable. (That is, if the host operating system is told the location of a block on an RN file, it can return the block without reading the entire file.) With RN files, it is not necessary to reach each record on the file to find a record of interest (as with sequential file). We tell the file manipulation routine where the block with the desired record is located and the requested block and record are returned. The catch is the location of the block must be remembered, and we use keys to do this.

For the simplest type of random file, we use record block pointers (RN, RTR's) as the keys to the blocks. These keys are primarily stored in the first block on the file which we call the index directory block (IDB). To locate a block, we tell the file manipulation routines to use the key in a particular word of the IDB. To actually access a particular block, the RN PTR to be used is indicated by the RN file manipulation routines, and that block will be returned to the program.

Since the keys to the data blocks have to be remembered if we are to directly and randomly access them, other kinds of blocks will be contained in the RN file. Every RN file has an index director block (IDB), and many files have other types of blocks used to store the keys. Data blocks on simple RN files will normally contain only one record. When this is true, the data blocks are the same size as the records, and they are variable length blocks. When several records are contained in a data block, then the routines that insert and extract records or data from the blocks must maintain the data block format.

Random files basically provide a low-level file manipulation method that performs disk block transfers from a program's main memory area to a mass

storage device. More sophisticated file manipulation routines use the RN files as the basic data structure over which they place a more complicated file organization. The symbolic keyed and numeric keyed files are examples of this.

Random files require more file space since they must store keys in addition to the records. Usually, this extra space is very minimal. The great advantage is that we are able to access any record directly. In the MC/DG data base, BASIS makes extensive use of random files.

(1ee) SYMBOLIC KEYED FILES

Many data processing applications require quick and easy access to records in a file. The records must be identified by a mnemonic key, and we must be able to directly access any record in the file at random. The programmer does not want to have to manage the keys or remember the location of the records. BASIS uses the Symbolic Keyed (SK) and Numeric Keyed (NK) file organizations to provide these capabilities.

Both SK and NK file manipulation routines (file access methods) provide for a machine independent method of creating and maintaining large files or randomly stored, variable size records that can be retrieved by mnemonic character string keys, by numeric keys, or by logical sequential position. The records in a file may vary in size from one character to a machine dependent maximum of 16,360 characters to 40,900 characters, depending on the host computer system. The minimum record length (MNRL) and maximum record length (MXRL) are defined by the programmer during file initialization. The variable length records are stored in fixed-length data blocks, usually each block will contain several records. A record space index (RSI) indicating the amount of available space in each data block is maintained within the file.

(1ff) OTHER FEATURES OF THE DATA BASE STRUCTURE

BASIS uses a non-hierarchical structure in its data bases. Every record can be related to any other record in the data base as the application requires because no hierarchical structure is unnecessarily forced on a data base. A number of different kinds of records may be in a data base, andthey can be related to one another in any logical way.

The data base in partitioned into as many as six different types of files. Each file is used to support some feature(s) available in BASIS. Since each file is maintained by the File Maintenance System in a somewhat different manner, a certain amount of parallel processing can be done. Once the input data is processed, the transactions generated to change the files of the data base can be handled at the same time by group. We are able to update the Source Data File and the Index File at the same time. Each set of index transaction types (index, range, and thesaurus) may be sorted and acted on in parallel processing.

Since a data base is partitioned into different files, each file may be used independently of the other files. Sometimes it is not economically feasible to store documents in a literature file on line, but it is necessary

to use an automated method to search for the documents. In this case, the data base may consist of only an Index File and the user would be shown accession numbers of documents that satisfy his retrieval criteria. The accession number would then be used to manually locate the document.

☆U.S.Government Printing Office: 1981 — 757-002/346

Recd
4/5/2001

DEPARTMENT OF THE AIR FORCE
AIR FORCE RESEARCH LABORATORY
WRIGHT-PATTERSON AIR FORCE BASE OHIO 45433

27 March 2001

MEMORANDUM FOR: Defense Technical Information Center/BCS
8725 John J. Kingman Rd, Suite 0944
Ft Belvoir, VA 22060-6218

FROM: Det 1 AFRL/WST (STINFO)
Bldg 640 Rm 60
2331 12th Street
Wright-Patterson AFB OH 45433-7950

SUBJECT: Notice of Change for Technical Reports

1. Reference AFWAL-TR-80-4115, Vol. 3; Title: ICAM Manufacturing Cost/Design Guide, Volume 3: Computerization; AD B206445; and AFWAL-TR-80-4115, Vol. 2, Title: ICAM Manufacturing Cost/Design Guide, Volume 2: Appendices to Demonstration Sections, AD B206444..

2. Please change distribution statement on these reports *from* Distribution Statement B (US Government agencies only; Test & Evaluation) *to* Distribution Statement A (Approved for public release, distribution unlimited).

3. Please call me at DSN 785-5197, if more information is needed.

Lolita V. Mitchell
LOLITA V. MITCHELL
STINFO and Tech Editing
Technical Information Branch